# IOWA STATE UNIVERSITY Digital Repository

Graduate Theses and Dissertations

Graduate College

2010

# Frequency Domain Active Noise Control with Ultrasonic Tracking

Thomas Chapin Waite Iowa State University

Follow this and additional works at: http://lib.dr.iastate.edu/etd Part of the <u>Mechanical Engineering Commons</u>

#### **Recommended** Citation

Waite, Thomas Chapin, "Frequency Domain Active Noise Control with Ultrasonic Tracking" (2010). *Graduate Theses and Dissertations*. 11782. http://lib.dr.iastate.edu/etd/11782

This Dissertation is brought to you for free and open access by the Graduate College at Iowa State University Digital Repository. It has been accepted for inclusion in Graduate Theses and Dissertations by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

### Frequency domain active noise control with ultrasonic tracking

by

Tom Waite

# A dissertation submitted to the graduate faculty in partial fulfillment of the requirements for the degree of DOCTOR OF PHILOSOPHY

Major: Mechanical Engineering

Program of Study Committee: Atul G. Kelkar, Major Professor Julie A. Dickerson J. Adin Mann III Jerald M. Vogel Qingze Zou

Iowa State University

Ames, Iowa

2010

Copyright © Tom Waite, 2010. All rights reserved.

# TABLE OF CONTENTS

LIST OF FIGURES	v
ACKNOWLEDGEMENTS	xi
ABSTRACT	хіі
CHAPTER 1. INTRODUCTION	1
1.1 Background	1
1.1.1 Frequency Domain ANC	5
1.1.2 Ultrasonic Tracking	7
1.2 Outline	8
CHAPTER 2. ULTRASONIC TRACKING	11
2.1 Introduction $\ldots$ $\ldots$ 1	11
2.2 Localization $\ldots$ 1	11
2.2.1 Bandwidth Considerations	16
2.3 Error $\ldots$ $\ldots$ 1	18
2.3.1 Sources of Incorrect Solutions	19
2.3.2 Sources of Error	20
2.3.3 Error w.r.t. Microphone Position	24
2.4 Experimental Results	27
2.5 Remarks	32
CHAPTER 3. NON-MINIMUM PHASE DECOMPOSITION AND	
SYSTEM IDENTIFICATION	33

3.1	Introduction	33
3.2	The Time Delay Problem	33
3.3	Non-minimum Phase System Decomposition	38
3.4	System Identification	40
	3.4.1 Obtaining the Frequency Response	41
	3.4.2 Experimental Results	44
3.5	Remarks	50
CHAP	TER 4. TIME ADVANCING WITH THE STFT	52
4.1	Introduction	52
4.2	Controller as Time Advance Filter	52
4.3	Signal Deconstruction and Reconstruction	54
4.4	STFT Time Advancement	59
4.5	STFT Filter Performance	61
4.6	Remarks	66
4.6 CHAP	Remarks       TER 5. TIME ADVANCING WITH THE PHASE VOCODER	66 67
4.6 <b>CHAP</b> 5.1	Remarks	66 67 67
4.6 <b>CHAP</b> 5.1 5.2	Remarks	66 67 67 67
4.6 <b>CHAP</b> 5.1 5.2	Remarks	66 67 67 67 68
4.6 <b>CHAP</b> 5.1 5.2	Remarks       TER 5. TIME ADVANCING WITH THE PHASE VOCODER         Introduction       Introduction         The Phase Vocoder       Introduction         5.2.1       PV Frequency Estimation         5.2.2       Performance	66 67 67 67 68 70
4.6 <b>CHAP</b> 5.1 5.2 5.3	Remarks	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> </ul>
4.6 CHAP 5.1 5.2 5.3	Remarks	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> <li>75</li> </ul>
4.6 <b>CHAP</b> 5.1 5.2 5.3	Remarks	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> <li>75</li> <li>80</li> </ul>
4.6 <b>CHAP</b> 5.1 5.2 5.3 5.3	Remarks   TER 5. TIME ADVANCING WITH THE PHASE VOCODER   Introduction   Introduction   The Phase Vocoder   5.2.1   PV Frequency Estimation   5.2.2   Performance   5.3.1   Performance   5.3.2   Error   Remarks	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> <li>75</li> <li>80</li> <li>85</li> </ul>
4.6 CHAP 5.1 5.2 5.3 5.4 CHAP	Remarks   TER 5. TIME ADVANCING WITH THE PHASE VOCODER   Introduction   The Phase Vocoder   The Phase Vocoder   5.2.1 PV Frequency Estimation   5.2.2 Performance   PV Time Advance Filter   5.3.1 Performance   5.3.2 Error   Remarks   TER 6. INTERNAL MODEL CONTROL	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> <li>75</li> <li>80</li> <li>85</li> <li>86</li> </ul>
4.6 CHAP 5.1 5.2 5.3 5.4 CHAP 6.1	Remarks	<ul> <li>66</li> <li>67</li> <li>67</li> <li>68</li> <li>70</li> <li>73</li> <li>75</li> <li>80</li> <li>85</li> <li>86</li> <li>86</li> </ul>

6.3	Interna	al Model Control	•	 •	•			•	89
	6.3.1	Periodic Signal Selector	•		•				91
	6.3.2	Stability	•		•				96
6.4	Experi	mental Results	•	 •	•			•	104
6.5	Remar	ks	•	 •					120
CHAP	TER 7	CONCLUSION AND FUTURE WORK .	•	 •	•	•	• •		121
7.1	Conclu	sion	•	 •	•				121
7.2	Future	Work	•	 •					123
BIBLI	OGRA	РНҮ				•	• •		127

# LIST OF FIGURES

Figure 1.1	Active noise control performance review under various conditions.	3
Figure 1.2	Diagram showing interaction of tracking system and controller	4
Figure 1.3	Tractor cab example.	5
Figure 1.4	Comparison chart of positioning systems	8
Figure 2.1	Each speaker to microphone acoustic delay appears as a peak on	
	the cross-correlation plot	13
Figure 2.2	Speaker to microphone distances can be extracted from the slope	
	of the cross-spectrums	14
Figure 2.3	Ultrasonic localization of a target within a 3D environment	15
Figure 2.4	Effect of bandwidth on the cross-correlation.	17
Figure 2.5	Time separated pulses can each utilize the entire available band-	
	width.	17
Figure 2.6	Intersecting additive error margins are shown at various distances	
	to illustrate potential microphone positions	23
Figure 2.7	Multiplicative error gets worse with distance	23
Figure 2.8	Blown up view of error margins intersecting to form a region of	
	possible microphone locations.	24
Figure 2.9	Maximum error as a function of microphone position in a $2D$	
	environment. Speakers located at $(0,0)$ and $(1,0)$	26
Figure 2.10	Block diagram of 3D ultrasonic tracking system	28

Figure 2.11	Ultrasonic speaker array and microphone to be tracked	29
Figure 2.12	Tracking a microphone as it follows a circular trajectory results	
	in $\pm 0.5$ cm error	30
Figure 2.13	Tracking a distant circular path results in $\pm 1$ cm error	30
Figure 2.14	Measured position captures microphone movement behind an ob-	
	struction by detecting diffracted acoustic waves. $\ldots$	31
Figure 3.1	Collocated, minimum phase systems show phase responses which	
	are much easier to control than their non-collocated counterpart.	35
Figure 3.2	A 3D acoustic system model separated into its minimum phase	
	part and its time delay (modeled using an all-pass filter approx-	
	imation). As is often the case, more poles and zeros are needed	
	to represent the time delay than the acoustic dynamics	36
Figure 3.3	Root locus showing instability results with very little controller	
	gain due to the RHP zeros pulling the poles into the RHP. The	
	RHP zeros are a byproduct of the time delay	37
Figure 3.4	Block diagram for viewing real-time acoustic frequency response	
	data	43
Figure 3.5	Enclosure made to mimic a large tractor cab with the controller	
	workstation in the foreground	45
Figure 3.6	Interior of the enclosure showing the control speaker and micro-	
	phone	45
Figure 3.7	The frequency response of the tractor cab is extremely complex	
	with large phase lag.	48
Figure 3.8	The 80 state model of the enclosure frequency response has an	
	average magnitude error of only $0.217~\mathrm{dB}$ and an average phase	
	error of only $1.79^{\circ}$ from 30-800 Hz	49

Figure 3.9	The model of the minimum-phase acoustic system shows no right	
	half plane zeros and has an equal number of poles and zeros so	
	it is stably invertible	50
Figure 4.1	Block diagram of closed loop feedback system showing decom-	
	posed plant, controller H(s), and time advance filter $e^{t_d s}$	54
Figure 4.2	Illustration of the overlap-add reconstruction of a signal which	
	has been deconstructed into several windowed segments	55
Figure 4.3	Progression illustrating a window's maximum hop size parameter	
	as well as its role in signal scaling	57
Figure 4.4	Comparison of window parameters important to signal recon-	
	struction.	58
Figure 4.5	Multiplication with $e^{j\omega t_d}$ linearly increases phase without chang-	
	ing magnitude	59
Figure 4.6	STFT time advancement block diagram with example signal to	
	illustrate each step	60
Figure 4.7	The STFT filter perfectly reconstructs a time advanced 32 Hz	
	sine wave.	61
Figure 4.8	The STFT filter incorrectly reconstructs a time advanced 34 Hz	
	sine wave.	62
Figure 4.9	Increasing the buffer time increases the number of accurately time	
	advanced frequencies. No matter how large the buffer, though,	
	there will always be large error in the spaces between center fre-	
	quency bins of the FFT	65
Figure 5.1	Comparing successive phases yields a different frequency estimate	
	for each n (see Eq. 5.1)	69

Figure 5.2	With a short preview window, STFT frequency estimation is	
	coarse while the phase vocoder is able to follow the chirp to within	
	0.1 Hz	71
Figure 5.3	Frequency estimation vs preview time (window size) using the	
	STFT and PV methods.	72
Figure 5.4	Block diagram of time advance filter using phase vocoder	75
Figure 5.5	Time advance filter with the phase vocoder accurately reproduces	
	previously inaccurate 34 Hz tone	76
Figure 5.6	Multi-tone PV filter error in time domain. Signal is analyzed in	
	first portion and canceled in second	78
Figure 5.7	Multi-tone PV filter error in frequency domain shows accuracy	
	at tones but no where else. Magnitude plot also shows input and	
	output of filter.	79
Figure 5.8	PV time advance filter error versus tonal separation for several	
	window sizes	81
Figure 5.9	PV time advance filter error versus chirp speed for several window	
	sizes	82
Figure 5.10	PV time advance filter error versus tonal separation for each win-	
	dow type	84
Figure 5.11	PV time advance filter error versus chirp speed for each window	
	type	84
Figure 6.1	Feedback control block diagram	87
Figure 6.2	Standard feedback configuration yields oscillation in performance	
	due to control delays.	88
Figure 6.3	Internal model control block diagram.	90

Figure 6.4	IMC solves control oscillation problem and reduces noise of $300$	
	Hz test signal substantially	1
Figure 6.5	Using the IMC, tonal noise reduction is good, but surrounding	
	broadband noise is increased	2
Figure 6.6	The STFT magnitude vector of the time advance filter is trun-	
	cated by the periodic signal selector to lessen random noise effects	
	on performance	4
Figure 6.7	Using the periodic signal selector within the STFT allows for	
	tonal noise reduction without increasing broadband noise 9	5
Figure 6.8	Region of instability shown as a function of time advance filter	
	error (y-axis) and tracking error (x-axis)	2
Figure 6.9	Region of stability diminishes with increasing error between mod-	
	eled and actual plant	3
Figure 6.10	Spectrogram of water pump	8
Figure 6.11	Spectrogram of water pump with IMC	8
Figure 6.12	Spectrogram of desk fan	9
Figure 6.13	Spectrogram of desk fan with IMC	9
Figure 6.14	Spectrogram of bathroom fan	0
Figure 6.15	Spectrogram of bathroom fan with IMC	0
Figure 6.16	Spectrogram of bus	1
Figure 6.17	Spectrogram of bus with IMC	1
Figure 6.18	Spectrogram of Bobcat idling	2
Figure 6.19	Spectrogram of Bobcat idling with IMC	2
Figure 6.20	Spectrogram of diesel engine	3
Figure 6.21	Spectrogram of diesel engine with IMC	3
Figure 6.22	Spectrogram of large boat	4
Figure 6.23	Spectrogram of large boat with IMC	4

Figure 6.24	Spectrogram of ship cabin
Figure 6.25	Spectrogram of ship cabin with IMC
Figure 6.26	Spectrogram of car driving
Figure 6.27	Spectrogram of car driving with IMC
Figure 6.28	Spectrogram of car getting onto highway
Figure 6.29	Spectrogram of car getting onto highway with IMC 117
Figure 6.30	Spectrogram of compressor
Figure 6.31	Spectrogram of compressor with IMC
Figure 6.32	Spectrogram of small propeller airplane
Figure 6.33	Spectrogram of small propeller airplane with IMC
Figure 7.1	The plant selector is the link between the tracking system and
	controller
Figure 7.2	Possible feedforward/feedback integrated controller
Figure 7.3	STFT computation blocks of a multi-resolution system 126

# ACKNOWLEDGEMENTS

I would like to thank Dr. Atul Kelkar for his guidance and support throughout this project. I also want to thank Dr. Julie Dickerson, Dr. Adin Mann, Dr. Jerald Vogel, and Dr. Qingze Zou for serving on my program of study committee. This dissertation is dedicated to my wife and family for their love and support.

## ABSTRACT

In this work, a new type of feedback control is presented which operates primarily in the frequency domain for use in 3D active noise control problems. Within the controller, a so-called time advance filter is designed to overcome the delay caused by the acoustics and the FFT. The filter is able to manipulate the phase response of the input so that the output appears time advanced. This nonlinear frequency domain feedback control method is shown to have more versatility, improved stability, and significantly reduced waterbed effect as compared to linear feedback control methods. An ultrasonic tracking system is also presented as a way to identify the location of a person's ear in real-time within a 3D environment. This position information is then provided to the controller which can adapt based on the user's position to achieve improved performance. Although ultrasonic tracking of a microphone is not new, this is the first time its potential for use in active noise control is shown.

### CHAPTER 1. INTRODUCTION

#### 1.1 Background

Well-sealed headsets lined with sound absorbing foam have been the traditional solution for mitigating high noise levels. This passive control approach has shown very large broadband reductions of noise. However, in some cases, people working in a noisy environment do not wear headsets because they can be uncomfortable to wear for long periods of time. An active noise control system would allow these people to be free of an uncomfortable headset while reducing their risk of hearing loss. Also, passive headsets reduce desired noise such as speech and music, whereas active systems have the ability to selectively reduce only undesired noise. These are a few of the motivations behind the research presented in this work.

Active headsets attempt to improve low frequency (< 1 kHz) noise reductions by electronic means. An active headset can be more comfortable than a passive headset because it does not require a large clamping force around the ears to achieve its performance. It is also quite simple to develop since a headset can be modeled as a 0D (zero dimensional) acoustic problem. This is due to the controller microphone and speaker being in such close proximity to the user's ear, enabling acoustic dynamics to be ignored since this separation is significantly smaller than the wavelength of the highest frequency being controlled (34 cm at 1 kHz). By ignoring the dynamics, an analog feedforward noise controller can be designed simply using an inverting amplifier and a low pass filter. Although this type of controller achieves excellent broadband low frequency noise reductions, it still requires the user to wear a headset.

Active noise control can also be implemented without a headset using speakers and microphones positioned near the user. The acoustics within a 3D environment are extremely complex even for simple room geometries, making modeling the dynamics of the system very difficult. In addition, the acoustics can change significantly as objects or people enter or leave the room. Successful active noise control in this scenario typically requires the use of a powerful digital signal processor (DSP) to act as controller. This 3D noise control is the current focus of much research as it has potential to reduce noise in industrial or transportation settings where passive means are impractical due to size and weight constraints.

At this point it is helpful to present a review of active noise control capabilities, as shown in Fig. 1.1. This table breaks down the noise control problem into two acoustic environments and shows the performance capabilities of both feedforward and feedback control under various conditions within these environments. It can be seen that within the 0D/1D acoustic scenario of a headset or duct problem, performance is good for a variety of conditions. This is due to the simple acoustics involved with headsets and ducts and is why these were the first commercializations of the technology. Headsets have primarily used analog feedforward control while duct solutions have implemented digital adaptive feedforward control which performs better when acoustics change with time. Feedback control is typically not used as the primary control mechanism because it is not able to cancel transient noise and has a more limited bandwidth when dealing with broadband noise.

Performance capabilities diminish in several respects when 3D noise control of a room or outdoor environment is considered. From Fig. 1.1 it can be seen that broadband noise can no longer be controlled well using feedback. This is because the time delay between

	0D/1D Acoustics 2D/3D Acoustics				tics		
	(Headset,	Duct)	(R	oom, Outdo	ors)		
	Feedforward	Feedback	k Feedforward Feedback New Syst				
Narrowband/Tonal	✓	✓	✓	1	✓		
Broadband	✓	✓	✓	х	х		
Moving Source	✓	✓	*	✓	✓		
Moving User	✓	✓	*	*	✓		
Transients	✓	х	✓	х	х		
Changing Acoustics	x	✓	x 🗸 🗸				

- Good Performance
- \* Moderate performance using multiple speakers and mics
- x Poor Performance

Figure 1.1 Active noise control performance review under various conditions.

the control speaker and error microphone is typically larger than the wavelengths under consideration, which limits the controller to periodic (tonal) disturbances. Also, performance of feedforward control diminishes in 3D problems when the noise source is moving or the user is moving. This can be improved somewhat using multiple speakers and microphones connected to a more advanced multi-input/multi-output (MIMO) controller. Feedback control performance is immune to noise source movement, but performs poorly when the user moves. This is the problem the new tracking system attempts to eliminate when used in conjunction with the frequency domain feedback controller.

Much of the previous work in active noise control utilizes the Least Mean Squares (LMS) routine or one of its many variations to update feedforward filter coefficients [1]. The LMS algorithm continuously compares the error microphone signal with a reference signal to find the optimal filter which results in the least error (at least in the quadratic sense). The optimal coefficients are found simply by solving a linear set of equations.



Figure 1.2 Diagram showing interaction of tracking system and controller.

This adaptive filter is usually implemented as an FIR filter because it is inherently stable and the LMS algorithm is guaranteed to converge to a global minimum. Since an IIR filter introduces poles into the system, there is a danger of the LMS algorithm converging to an unstable solution or a local minimum and is therefore rarely used in practice [2].

In this research an IIR filter was used because the plant model was assumed accurate and unchanging and therefore LMS adaptation would not be needed. Real world implementation may necessitate the use of LMS adaptation, although since this research also introduces an accurate tracking system, it is suggested to use the error microphone position data to adapt the plant model. In the design stage, the engineer could experimentally identify a multitude of plant models at known locations within the room. This would create a 3D acoustic map of the room which would be unique to each environment. In this way, all of the potential models could be stored in memory and the adaptation algorithm would only be needed to switch and/or interpolate between them. Such a design is illustrated in Fig. 1.2.



Figure 1.3 Tractor cab example.

A good example setting for this type of position-based adaptive control is a tractor. This is because the room geometry is small and stays primarily fixed with time, the user is sitting and therefore not a fast moving target, and the disturbance noise is loud and primarily low in frequency. Possible speaker and microphone locations are shown in Fig. 1.3. The error microphone needs to be attached to the person near the ear, perhaps on sunglasses, a hat, or a shirt collar. Ideally the microphone would be a part of an earplug inserted in the ear which would transmit wirelessly to the controller.

#### 1.1.1 Frequency Domain ANC

The use of transforms such as the FFT are rare in active noise control because of the relatively large time delay they introduce into the system. Although it may be computationally efficient to use the FFT, it generally requires a large number of buffered samples for meaningful use. The time delay introduced by this buffer has been considered too large for many real-time applications such as active noise control. It should be noted that the FFT has been used in active noise control, but only as a means of periodically updating a feedforward plant model [3][4]. In this work, the FFT will be used as a frequency domain filter, directly applied to the error signal in the feedback path. This type of frequency domain filtering has not been done in active noise control and is the major contribution of this research.

The manner in which the FFT filter operates in the feedback path is by altering the gain and phase of each frequency identified by the transform. The delay and gain adjustment vectors are determined by the controller such that when the IFFT is taken and the signal is output by the speaker it will cancel any periodic disturbance. A method of using a single variable delay and gain to cancel a single frequency has been previously shown [5]. However, since this new method uses a vector of gain and phase values corresponding to the length of the FFT, multiple frequencies can be canceled.

As will be seen later, the FFT filter by itself cannot estimate frequencies accurately enough to adjust the phase of the signal the appropriate amount. However, by comparing the phase vectors of successive FFT's, frequency estimation accuracy dramatically improves. This is the main principle behind a phase vocoder. Although it is commonly used to shrink or expand a signal in time without changing its pitch, the phase vocoder has also been shown in previous work to estimate frequency components of a signal quickly and accurately [6]. It is this trait that makes its use desirable for frequency domain noise control. This is the first use of phase vocoder principles in active noise control.

#### 1.1.2 Ultrasonic Tracking

Motivation for user tracking for use in active noise control stems from the disparity in performance between local and global control. Local control typically involves a single control speaker and error microphone to reduce noise locally (i.e. in a small region surrounding the microphone). Performance has been shown to be very good as long as the user's ear is very close to the error microphone [7][8]. Since it is generally impractical to assume a person, even when seated, could keep their head position close to the microphone at all times, global noise control was developed. As the name implies, global noise control attempts to reduce noise not just in the vicinity of one error microphone, but of an entire room. Using more complex control and multiple speakers and microphones, global noise control has been shown to achieve moderate reductions at many locations within a room [9][10]. By tracking the user's ear position, a simple local controller can adapt based on this position, in effect achieving global noise control while maintaining the great performance of local control.

There are several tracking technologies available [11]-[14]. A table of some of these technologies has been given for comparison in Fig. 1.4. Looking at the accuracy of GPS and RF tracking, one can see they are used for large scale tracking (RF being an indoor substitute for GPS) and therefore not good candidates for the precision tracking required in this research. Both the infrared and ultrasonic tracking systems have similar accuracy as well as cost. Due to the speed of sound, the update rate of ultrasonic tracking is somewhat limited, reducing the accuracy as the speed of the target increases. IR tracking does not have this problem, as its update rate is only limited by the quality of the camera. However, the IR tracking system requires an infrared LED or reflector be placed on the user. This is to the ultrasonic system's advantage since it needs only

	Infrared	Ultrasonic	Radio Frequency	GPS
Hardware Required	2 IR cameras,	3 speakers,	3 RF transmitters,	1 receiver
	1 IR LED	1 microphone	1 RF receiver	
Approx. Hardware Cost	\$100	\$100	\$200	\$200
Approx. Accuracy	1 cm	1 cm	1 m	1 m
Tracking Speed	>100 Hz	<50 Hz	1-10 Hz	1-10 Hz
Additional Hardware on User	IR LED	None	RF receiver	GPS receiver
Line of Sight?	Yes	No	No	Yes

Figure 1.4 Comparison chart of positioning systems.

the microphone which is already on the user (the error microphone used for feedback noise control). Another drawback of IR tracking is that it requires line of sight between the cameras and LED to function. As will be shown in this research, ultrasonic tracking is robust to path obstructions. Given all of this information, ultrasonic tracking was chosen as the best technology to use.

The localization of a noise source using several microphones has been shown in previous work and has many applications, from improving robot binaural hearing to simply determining the root cause of a disturbing noise [15][16]. Ultrasonic tracking using several speakers to track a microphone has less immediate applications but has also been shown in previous work [11][12]. In the research presented here, an extension of this work is given for the purpose of tracking a person's ear position, thereby enabling adaptive localized control which would improve performance of an active noise controller.

### 1.2 Outline

In chapter 2, a system using ultrasonic speakers for tracking a microphone's position as it moves through a 3D environment is presented. Microphone position information would improve performance of an active noise control system by providing adaptive localized control. Localization will be achieved by triangulating three speaker to microphone distances computed using the cross-correlation method. Sources of error as well as causes of incorrect solutions will be identified and analyzed. Experimental results of the real-time tracking system will be given using a simple path trajectory.

Chapter 3's primary focus will be on the detrimental effects of acoustic time lag inherent to 3D active noise control environments. The fundamental time delay problem will be presented with emphasis on the compromise between controller bandwidth and controller performance. Several means of decomposing a system into its minimum and non-minimum phase portions will be shown as a way of extracting the time delay information. An algorithm for computing and viewing frequency responses in real-time will also be shown. Using a small room enclosure as an example system, a frequency response is taken and decomposed to eliminate the effect of the time delay. The resulting minimum phase response is modeled using system identification software.

In chapter 4, a so-called time advance filter is introduced which uses the STFT to shift the phase of a signal's multiple frequency components the precise amount needed to appear time advanced. It is the hope that using this filter will negate the ill effects of the time delay which greatly restrict controller efficacy. The procedure used to deconstruct a signal into small segments using the STFT and then reconstruct it back using the ISTFT is given. This is needed since the time advance filter acts in the frequency domain. Several simulation results are given which show the performance and limitations of the time advance filter.

In chapter 5, a phase vocoder will be used to provide quick frequency and phase estimates of an incoming signal. This will greatly improve functionality of the time advance filter presented in the previous chapter. Specifically, it will allow for accurate time advancing not just at FFT frequency bins but at all frequencies up to the Nyquist rate. Several test signals will be used to evaluate the performance of the filter. Sources of error will also be presented and quantified to help the reader understand the limitations of the algorithm. The impact on performance of various parameters such as window size and type will also be discussed.

Chapter 6 will demonstrate active noise control performance using the previously developed time advance filter. Both a standard feedback controller and an internal model controller typically used for systems with delays will be investigated. Controller performance will be improved with the implementation of a periodic signal selector which separates periodic noise from random noise in real-time. An analysis of the internal model controller stability will be given with conditions derived from the Nyquist stability criteria. Closed loop experimental results of the control system using several example noises will be shown using spectrograms of both the uncontrolled and controlled signals.

# CHAPTER 2. ULTRASONIC TRACKING

### 2.1 Introduction

A system using ultrasonic speakers for tracking a microphone's position as it moves through a 3D environment is presented in this chapter. Microphone position information would improve performance of an active noise control system by providing adaptive localized control. Localization will be achieved by triangulating three speaker to microphone distances computed using the cross-correlation method. Sources of error as well as causes of incorrect solutions will be identified and analyzed. Experimental results of the real-time tracking system will be given using a simple path trajectory.

### 2.2 Localization

In acoustics problems, sources of noise are usually the target of localization. Finding from where a noise source is emanating generally requires several microphones positioned around the source. The microphones can then synchronously capture noise and comparison of the acoustic time delays between microphones provide the information needed to find the location of the source.

In this chapter, the focus will be switched so that instead of locating a source using several microphones, several sources (speakers) will be used to locate a microphone. The noise used will be ultrasonic so the tracking will take place without interfering with the control or disturbing the user. This localization routine will run continuously to enable real-time tracking of the microphone. With the microphone positioned close to the user's ear, adaptive localized noise control can be used to achieve excellent performance even as the user moves.

In order to determine the location of a microphone uniquely in 3D space, four speakers are needed. Also, the four speakers must encompass a volume. This means that if one were to draw an invisible line between each speaker, the lines would form edges of a tetrahedron. In many cases, however, the solution space can be cut in half if one assumes the microphone will always be positioned in front of the speakers. If this is the case, only three speakers are needed. The localization algorithm produces two possible answers, one of which can be ignored because it lies behind the speakers and therefore outside the solution space. In this situation, the three speakers must encompass an area. So if one were to draw an invisible line between each speaker, the lines would form edges of a triangle.

Bandlimited white noise in the ultrasonic frequency range can be generated for use by the speakers. Each speaker could utilize 1/3 of the available bandwidth so that the microphone could easily distinguish between them. For example, if the usable bandwidth was 20-29 kHz, one speaker would play bandlimited white noise from 20-23 kHz, another from 23-26 kHz, and the last from 26-29 kHz. The usable bandwidth is limited on the lower end by human hearing (to keep it ultrasonic, it must be above 20 kHz) and limited on the upper end by speaker and microphone frequency responses. It is therefore critical to get speakers capable of playing and microphones capable of sensing high ultrasonic frequencies to maximize bandwidth.

The ultrasonic signals can be played and recorded synchronously so that the acoustic time delays between each speaker and the microphone can be determined. Using the



Figure 2.1 Each speaker to microphone acoustic delay appears as a peak on the cross-correlation plot.

cross-correlation method of Eq. 2.1, one can find the time delay unique to each speaker to microphone distance. The output of all three cross-correlations are plotted in Fig. 2.1. In each plot, the time at which the maximum value occurs corresponds to when the two signals (speaker and microphone) match each other the most. Since this occurs when the speaker signal is delayed by the acoustic time delay, these maximums coincide with the desired acoustic time delays.

Cross Correlation = 
$$F^{-1}[F(spk). * conj(F(mic))]$$
 (2.1)

where F is the FFT,  $F^{-1}$  the IFFT



Figure 2.2 Speaker to microphone distances can be extracted from the slope of the cross-spectrums.

The acoustic time delays can also be found with a similar method using the cross-spectrums of the microphone and speaker signals. From Eq. 2.2, one can see that the cross-spectrum is essentially the cross-correlation before it is transformed back into the time domain. Fig. 2.2 plots the three frequency domain cross-spectrums. The desired time delays can be found by calculating the slopes of the unwrapped phase responses. This frequency domain method is quite a bit faster than the time domain method since the IFFT does not need to be computed.

$$Cross Spectrum = F(spk) \cdot * conj(F(mic))$$
(2.2)

Once the three acoustic time delays have been computed, one just needs to compute  $d_i = ct_i$ , where c is the speed of sound,  $d_i$  is the distance and  $t_i$  is the acoustic time



Figure 2.3 Ultrasonic localization of a target within a 3D environment.

delay between the *i*th speaker and microphone. Combining these three distances with the speaker orientation information, cartesian coordinates for the microphone position can be calculated. To simplify the calculation, it is best if the speakers are oriented perpendicular to each other as shown in Fig. 2.3. With knowledge of speaker separations (a, b) and distances from speakers to the microphone  $(d_1, d_2, d_3)$ , one can calculate the microphone coordinates  $(x_m, y_m, z_m)$ . These quantities are all related by the pythagorean theorem shown in Eq. 2.3.

$$\begin{cases} x_m^2 + y_m^2 + z_m^2 = d_1^2 \\ (a - x_m)^2 + y_m^2 + z_m^2 = d_2^2 \\ x_m^2 + y_m^2 + (b - z_m)^2 = d_3^2 \end{cases}$$
(2.3)

In Eq. 2.4, the desired microphone coordinates are solved for in terms of the known quantities. Notice the equation for  $y_m$  is in terms of  $x_m$  and  $z_m$  to simplify the solution.

Also, the original solution for  $y_m$  included a  $\pm$  in front of the square root, indicating two possible microphone locations. However, as mentioned earlier, the solution space has been restricted to only microphone locations in front of the speakers to guarantee the uniqueness of the solution. Therefore, the positive solution is taken.

$$\begin{cases} x_m = \frac{1}{2a}(a^2 + d_1^2 - d_2^2) \\ z_m = \frac{1}{2b}(b^2 + d_1^2 - d_3^2) \\ y_m = \sqrt{d_1^2 - x_m^2 - z_m^2} \end{cases}$$
(2.4)

#### 2.2.1 Bandwidth Considerations

In this section, a closer look at the role bandwidth plays in the ultrasonic tracking system is provided. To determine the effect on performance, two signals with different size bandwidths are cross-correlated with time delayed versions of themselves in Fig. 2.4. The first signal uses bandwidth from 20-23 kHz while the second uses from 20-29 kHz. The 9 kHz bandwidth signal's cross-correlation reveals a much more narrow response with peaks much more rapidly decreasing in amplitude from the maximum peak. Since the maximum is what is searched for when determining the acoustic time delay, it is clear that using a larger bandwidth would be beneficial.

In a noisy real-world environment, the chance of detecting a false maximum is decreased with larger bandwidth. It is therefore important to utilize the maximum bandwidth available. As explained earlier, the usable bandwidth of the system is limited by the frequency response of the microphone and speakers. Using a correlation signal of bandlimited white noise, the speakers were restricted to 1/3 the available bandwidth because they played continuously and concurrently. Perhaps instead of using bandlimited white noise, bandlimited pulses could be used as correlation signals. If the pulses were



Figure 2.4 Effect of bandwidth on the cross-correlation.



Figure 2.5 Time separated pulses can each utilize the entire available bandwidth.

### 17

separated in time as in Fig. 2.5, they could each utilize the entire available bandwidth.

Using bandlimited pulses instead of bandlimited white noise improves the margin of error considerably. It also reduced computational cost considerably since now only one cross-correlation needs to be computed instead of three. The only extra cost of emitting three identical pulses is that once they've been recorded by the microphone, one must then be able to differentiate between them to determine which pulse came from which speaker. This can be done fairly easily by limiting the search region for the cross-correlation peak to just the region where it is expected. As we will see in the next section, this limiting of the search area also has the added benefit of reducing the likelihood of detecting of a false peak.

When designing such a system, caution must be taken so that the pulses are spaced far enough apart in time so they never overlap at the microphone. With knowledge of the environmental boundaries limiting the solution space, one can chose an appropriate pulse spacing without fear of overlapping the signals.

#### 2.3 Error

The analysis of error is obviously very important in determining a tracking system's feasibility. In this section, error is separated into two categories. One type of error is that of an incorrect solution, meaning the tracking algorithm chooses the wrong solution. In the other type of error, the tracking algorithm chooses the correct solution, but the difference between measured and actual parameters leads to error. Several causes for both these types of error will be presented. Potential fixes will also be given so that the tracking system can be designed to limit the influence of the error.

#### 2.3.1 Sources of Incorrect Solutions

- 1. Reflections off walls can correlate just as much as the direct acoustic paths. If a secondary path turned out to be the maximum peak in the cross-correlation, it would register a larger acoustic time delay and give an incorrect position. To reduce the likelihood of a reflection being misinterpreted as the direct path, measures could be taken to reduce reflections by adding acoustically absorptive materials to the walls. Since in many environments this is unfeasible, the tracking system can be programmed to electronically ignore the reflections. This is done by establishing the environmentally determined solution space and limiting the search for maxima to this space. All solutions outside of the confined solution space would then be ignored.
- 2. High directivity of the speakers can be a major problem since the high frequencies of ultrasound are extremely directional. If most of the acoustic energy is directed in front of the speakers and not dispersed to the sides, the low amplitude signal would be more susceptible to noise. This could create more false solutions in microphone positions which are off to the side of the speakers. One way to reduce the negative effect of directivity of the speakers is use wide-dispersion speakers. Another is to physically angle the speakers toward the most likely microphone position rather than positioning them flush with the wall. Yet another is to increase the number of ultrasonic speakers, intelligently placing them to increase coverage.
- 3. **Obstructions in the acoustic path** reduce the amplitude of the cross-correlation peak. The signal remaining consists of diffractions from the primary path and reflections from secondary paths. The acoustic tracking system will remain accurate for small obstructions. However, if the obstruction provides a significant enough

acoustic barrier, it could potentially reduce the peak to the level of the surrounding noise. In this case, a false solution can be avoided by increasing the sound power of the pulses so that the remaining diffracted signal is large enough to be detected in the noise. One could also position redundant sources at various locations and switch between them based on which has the largest cross-correlation peak.

- 4. Ultrasonic noise in the environment would interfere with the cross-correlation and can cause a false reading. This is usually not a concern since the large amount of ultrasonic noise that would be needed to cause problems is not present in most environments. A potential solution would be to increase the sound power of the pulses to overcome such noises.
- 5. Limited bandwidth causes the cross-correlation to have several peaks near the maximum. The smaller the bandwidth, the larger these surrounding peaks get. With severe bandwidth restrictions, only a small amount of noise can cause one of these false peaks to be incorrectly deemed the maximum. To combat this problem, the tracking system must use speakers and microphones with frequency responses that stretch as far into the ultrasonic range as possible.

#### 2.3.2 Sources of Error

1. User movement causes error for real-time tracking due to the fundamental acoustic time lag as well the computation time required to triangulate the position. Even with just a small enclosed space  $(1 \times 1 \times 1 \text{ m})$  to cover, the update rate of the tracking system is limited to about 100 Hz. This translates to a tracking error of around 1 cm for a 1 m/s moving target. In a scenario where a user is walking within a large room, the update rate would need to be reduced to accommodate the larger distance and acoustic lag. Assuming a standard walking speed of 1.5 m/s, the tracking error would be closer to 10 cm. With this amount of error, active noise control performance would be significantly hampered. It is therefore recommended to restrict the use of real-time ultrasonic tracking to small rooms with somewhat stationary users such as a seated person in a tractor cab.

- 2. Discrete sampling poses a problem for all digital tracking systems because it forces the actual time delay to be rounded to the nearest discrete sampled time. The measured delay is then within a margin of error proportional to the sampling time. Sampling at a rate of 68 kHz yields ±0.25 cm error to the distance measurement. To reduce this error, one can increase the sampling rate. For instance, doubling the sampling rate cuts the error in half. However, this will increase the price of the tracking system as well as the computational cost.
- 3. Speaker placement must be measured to a high precision since the speaker separation parameters (a,b) from Fig. 2.3 directly influence the calculations for the microphone coordinates (Eq. 2.4). For example, if the measured speaker separation distance was 1 m but the actual distance was 0.99 m, the error would propagate through to the measured microphone position. To reduce this error, one must take careful measurements of exact speaker positions. This can be difficult, especially since the speakers are not point sources. It is also possible to introduce several known test points, measuring the microphone position at each using ultrasonic tracking. The speaker separation distances could then be calibrated based on the results.
- 4. **Temperature** greatly impacts the speed of sound and therefore greatly effects the measured time delays. Eq. 2.5 shows the approximate relationship between

the speed of sound (m/s) and air temperature (°C). From this equation we can determine that a change in temperature of just 3.4°C yields approximately 1% error in time delay measurements. It is therefore critical to the performance of the tracking system to include ambient air temperature as an input into the system so that the speed of sound can be periodically adjusted.

$$c = 331.3 + 0.606T \tag{2.5}$$

Discrete sampling contributes to what is known as additive error. This means that based on the resolution of this parameter, the measured microphone position will be within a set error margin around the actual microphone position. This error margin does not change with distance. Fig. 2.6 shows the effect of additive error on tracking accuracy by plotting an exaggerated  $\pm 10$  cm error at several distances. In the figure, the blue dots represent the speakers, the shaded regions represent the possible measured microphone positions and the intersection of the blue circles are the actual microphone positions. As can be seen, even though the margin of error surrounding each distance measurement is fixed, the region of possible microphone positions grows with distance from the speakers both in the horizontal and vertical directions.

Mistakes in estimating the temperature of the air causes large errors in ultrasonic tracking. This is because misjudging the speed of sound has a compounding effect with distance. This is known as multiplicative error and is exhibited by a margin of error that increases with distance from the source. Fig. 2.7 shows the effect of multiplicative error on tracking accuracy by plotting an exaggerated  $\pm 5\%$  error at several distances. As might be expected, the regions of potential microphone positions grows quite rapidly with distance in such a system. Note: speaker placement imprecision contains both



Figure 2.6 Intersecting additive error margins are shown at various distances to illustrate potential microphone positions.



Figure 2.7 Multiplicative error gets worse with distance.


Figure 2.8 Blown up view of error margins intersecting to form a region of possible microphone locations.

additive and multiplicative error.

#### 2.3.3 Error w.r.t. Microphone Position

Every tracking system contains some form of additive and multiplicative error. One can only hope to minimize it by identifying problem areas and designing the tracking system accordingly. As we've seen from Fig. 2.6 and 2.7 error is very dependent on the location of the user in relation to the speakers. In this section, a derivation will be provided which determines the maximum tracking error at all locations within a 2D environment given specific additive and multiplicative errors.

Fig. 2.8 shows an enlarged portion of one of the error plots from before. The maximum tracking error,  $\epsilon$ , is defined in Eq. 2.6. Therefore, finding an  $\epsilon$  for each microphone position (x,y) requires finding  $(x_i, y_i)$  in terms of known parameters like additive error  $(\epsilon_A)$ , multiplicative error  $(\epsilon_M)$ , speaker to microphone distances  $(d_1, d_2)$  and speaker separation distance (a). Note the additive error is given as a distance but the multiplicative error is given as a percent. For example,  $\epsilon_A$ =0.01 would be 1 cm error but  $\epsilon_M$ =0.01 would equate to 1% error.

$$\epsilon = \max(\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4) \tag{2.6}$$

where 
$$\epsilon_{i} = \sqrt{(x - x_{i})^{2} + (y - y_{i})^{2}}$$

If we define the speaker to microphone distances like before as  $d_1 = \sqrt{x^2 + y^2}$  and  $d_2 = \sqrt{(a-x)^2 + y^2}$ , we can write eight equations which link all of these parameters in Eq. 2.7.

$$x_{1}^{2} + y_{1}^{2} = \left(d_{1}(1 + \epsilon_{M}) + \epsilon_{A}\right)^{2}$$

$$(a - x_{1}^{2}) + y_{1}^{2} = \left(d_{2}(1 + \epsilon_{M}) + \epsilon_{A}\right)^{2}$$

$$x_{2}^{2} + y_{2}^{2} = \left(d_{1}(1 + \epsilon_{M}) + \epsilon_{A}\right)^{2}$$

$$(a - x_{2}^{2}) + y_{2}^{2} = \left(d_{2}(1 - \epsilon_{M}) - \epsilon_{A}\right)^{2}$$

$$x_{3}^{2} + y_{3}^{2} = \left(d_{1}(1 - \epsilon_{M}) - \epsilon_{A}\right)^{2}$$

$$(a - x_{3}^{2}) + y_{3}^{2} = \left(d_{2}(1 - \epsilon_{M}) - \epsilon_{A}\right)^{2}$$

$$x_{4}^{2} + y_{4}^{2} = \left(d_{1}(1 - \epsilon_{M}) - \epsilon_{A}\right)^{2}$$

$$(a - x_{4}^{2}) + y_{4}^{2} = \left(d_{2}(1 + \epsilon_{M}) + \epsilon_{A}\right)^{2}$$

With some manipulation of Eq. 2.7, one can solve for  $(x_i, y_i)$ . These coordinates, which are necessary for determining maximum error, are given below (Eq. 2.8).

$$\begin{cases} (x_1, y_1) = \left(\frac{\left(d_1(1+\epsilon_M)+\epsilon_A\right)^2 - \left(d_2(1+\epsilon_M)+\epsilon_A\right)^2 + a^2}{2a}, \sqrt{\left(d_1(1+\epsilon_M)+\epsilon_A\right)^2 - x_1^2}\right) \\ (x_2, y_2) = \left(\frac{\left(d_1(1+\epsilon_M)+\epsilon_A\right)^2 - \left(d_2(1-\epsilon_M)-\epsilon_A\right)^2 + a^2}{2a}, \sqrt{\left(d_1(1+\epsilon_M)+\epsilon_A\right)^2 - x_2^2}\right) \\ (x_3, y_3) = \left(\frac{\left(d_1(1-\epsilon_M)-\epsilon_A\right)^2 - \left(d_2(1-\epsilon_M)-\epsilon_A\right)^2 + a^2}{2a}, \sqrt{\left(d_1(1-\epsilon_M)-\epsilon_A\right)^2 - x_3^2}\right) \\ (x_4, y_4) = \left(\frac{\left(d_1(1-\epsilon_M)-\epsilon_A\right)^2 - \left(d_2(1+\epsilon_M)+\epsilon_A\right)^2 + a^2}{2a}, \sqrt{\left(d_1(1-\epsilon_M)-\epsilon_A\right)^2 - x_4^2}\right) \end{cases}$$
(2.8)

By plugging in these values into Eq. 2.6, we can determine the maximum error,  $\epsilon$ , at each (x,y) coordinate in the entire 2D space. The results of this are given as a colormap plot in Fig. 2.9. A very large amount of error can be seen to the sides of the speakers



Figure 2.9 Maximum error as a function of microphone position in a 2D environment. Speakers located at (0,0) and (1,0).

and the error gets progressively larger with distance. A region of low error is found in between the speakers with the minimum error point in between the two speakers,  $45^{\circ}$ off-axis. This region of low error remains despite changes in parameters a,  $\epsilon_A$ , and  $\epsilon_M$ . It is therefore critical from an error minimization standpoint to place the tracking speakers in such a way that the user's most likely position is within this low error region.

Although there are many potential sources of error involved with ultrasonic tracking, that does not mean that it is inaccurate or inferior to other forms of tracking. As we will see in the next section, ultrasonic tracking can be quite accurate.

# 2.4 Experimental Results

The tracking algorithm used is displayed as a flow diagram in Fig. 2.10 and was implemented in real-time using Simulink interfaced with a DSP board. In the algorithm, three successive bandlimited pulses are sent to the speakers as well as through a nominal minimum phase plant model. The model is used to simulate the propagation path from speaker to most likely microphone position. Passing the pulses through this simulated acoustic path makes for a more correlated noise with the pulses that pass through the actual plant and are recorded by the microphone.

The plant model must be minimum phase so that the time delay is not incorporated into it. Otherwise, the cross-correlation would output zero time delay. A method for obtaining a minimum phase plant model is given in the next chapter. Although not shown in the diagram, it is computationally more efficient to run the pulses through the nominal minimum phase plant off-line and save a new set of filtered pulses for use as input to the cross-correlation block.

The three pulses are then all at once cross-correlated with the microphone signal, which should show three peaks at times corresponding to the propagation delay between each speaker and the microphone. To prevent the occurrence of a secondary reflected pulse accidentally being recorded as a peak instead of the primary path pulse, the cross-correlation is range limited so all solutions outside the preset feasible solution space are ignored. The time at which each peak occurs is then recorded and transformed into a distance measurement. To reduce the error of this calculation, a temperature sensor may be used to provide a more accurate value of the speed of sound (see Eq. 2.5). The distance measurements are then transformed into cartesian coordinates via Eq. 2.4 before they are finally recorded and/or viewed using a 3D scope.



Figure 2.10 Block diagram of 3D ultrasonic tracking system.

The speakers used in this experiment are wide dispersion piezo tweeters capable of playing 3-30 kHz. They are oriented as shown in Fig. 2.11, with a horizonal separation of 84 cm and a vertical separation of 50 cm. The speakers are angled inward to help with directionality problems. Also shown in this figure is the microphone used in the experiment. It is an omnidirectional miniature electret type commonly used in hearing aids. Its extremely small size helps to extend its frequency response into the ultrasonic range. The small size also makes it easily wearable without causing discomfort or distraction.

For the experiments, a microphone was attached to a motorized horizontal disc in front of the speaker array. With the ultrasonic tracking system operating, the disc was rotated through several revolutions. This simple circular path provided an excellent way of testing the accuracy of the system since the measured microphone positions could be directly compared with the known trajectory. In each of the experiments, the microphone traverses 1.29 m/rev at a rate of 0.72 m/s. These values were used as an estimation of the maximum range and speed of a typical seated person. The sampling rate was 65536 Hz ( $2^{16}$ ), so a bandwidth of 20-32 kHz was used to generate the pulses.



Figure 2.11 Ultrasonic speaker array and microphone to be tracked.

The update rate of the positioning was 64 Hz.

The results of the first experiment are shown in Fig. 2.12. Here, the center of the rotating disk is placed in between the speakers in a region of expected low error. The tracking system is able to follow the microphone trajectory to within  $\pm 0.5$  cm. The worst error appears near small y-coordinates. This is expected from Fig. 2.9 since the circle's lower trajectory extends outside the region of low error.

For the next experiment, the circle was placed much further out from the speakers (approximately 1.5 m). The results of Fig. 2.13 shows an error margin of around  $\pm 1$  cm. The worst error appears along the x-axis. This was expected by referring to Fig. 2.6 or 2.7 and observing the large x-axis error compared to the y-axis error at far distances.

The final experiment shows the effect of placing an obstruction in between the speaker and the microphone in an attempt at simulating a real world situation. As shown in Fig. 2.14, a mannequin head was used to block the line of sight of the direct acoustic path.



Figure 2.12 Tracking a microphone as it follows a circular trajectory results in  $\pm 0.5$  cm error.



Figure 2.13 Tracking a distant circular path results in  $\pm 1$  cm error.



Figure 2.14 Measured position captures microphone movement behind an obstruction by detecting diffracted acoustic waves.

Although the error increased, the tracking system was still able to follow the microphone movement without a direct acoustic path. This is because a small portion of the acoustic wave was able to diffract around the head and still be detected by the microphone. Since the wave has to travel around the head, the recorded distance is greater than the direct path. This explains why the measured microphone position was mistakenly tracked further away from the speaker obstructed with the head. Although this introduces a slight error, it is much better than losing the position information altogether. The ability to track an object even without direct line of sight is a significant advantage for ultrasonic positioning.

# 2.5 Remarks

An ultrasonic positioning system was developed to track a microphone in 3D space. The microphone position information could be fed into an active noise control system to update internal models and improve performance. The use of continuous bandlimited white noise and time separated bandlimited pulses were both explored. Pulses were deemed superior due to the increased bandwidth available. Both time and frequency domain methods for determining acoustic time delays were given. Many sources of error and causes of incorrect solutions were identified as well as ideas for reducing the error and likelihood of an incorrect solution. Experimental results showed that ultrasonic tracking could be quite accurate for tracking situations involving slow moving targets in a confined space. Ultrasonic tracking also showed its ability to track a target positioned behind an obstruction, a key advantage over infrared tracking.

# CHAPTER 3. NON-MINIMUM PHASE DECOMPOSITION AND SYSTEM IDENTIFICATION

# 3.1 Introduction

This chapter's primary focus will be on the detrimental effects of acoustic time lag inherent to 3D active noise control environments. The fundamental time delay problem will be presented with emphasis on the compromise between controller bandwidth and controller performance. Several means of decomposing a system into its minimum and non-minimum phase portions will be shown as a way of extracting the time delay information. An algorithm for computing and viewing frequency responses in real-time will also be shown. Using a small room enclosure as an example system, a frequency response is taken and decomposed to eliminate the effect of the time delay. The resulting minimum phase response is modeled using system identification software.

### 3.2 The Time Delay Problem

Simple dynamic systems can generally be modeled as mass-spring-damper systems. Acoustic systems differ from these systems in two important ways. For one, the order of the system is much larger, with mathematical models sometimes reaching over 100 states. This makes manual tuning of an acoustic controller impractical, limiting control strategies to mostly automatic ones like LQG (Linear Quadratic Gaussian) or LMS (Least Mean Squared). It also increases the computation burden on the controller, which in turn increases the cost. Another more detrimental difference between acoustic and many mechanical systems is the relatively large time delay between actuator (speaker) and sensor (microphone). The speed of sound may seem instantaneous to our ears, but the lag in time it takes for a sound wave to reach us from a speaker just feet away dramatically limits the capabilities of a controller.

This time delay problem can be understood by viewing the phase plot of Fig 3.1. Here, a system with a collocated sensor/actuator (speaker and microphone in same physical location) is compared to a system with a non-collocated sensor/actuator (speaker and microphone separated by some distance). The red dotted curve is the collocated system and the blue curve is the non-collocated system. The phase difference is described by Eq. 3.1. From a gain margin viewpoint, the collocated phase response never drops below  $-180^{\circ}$  and therefore has infinite gain margin. On the other hand, the non-collocated phase response drops below  $-180^{\circ}$  almost immediately and crosses  $-180^{\circ} - 360^{\circ}n$  points a multitude of times, indicating either a small gain margin or more likely a negative gain margin. Therefore, a controller would have a hard time performing well since it would be on the verge of instability.

$$\angle G(s) = \angle G_{mp}(s) - \omega t_d \tag{3.1}$$

The collocated acoustic system is labeled  $G_{mp}(s)$  because it is minimum phase. A system is designated minimum phase if it's transfer function model contains no right half plane zeros. Many acoustic systems can be decomposed into a minimum phase portion and a time delay as shown in Eq. 3.2. The minimum phase part can be exactly



Figure 3.1 Collocated, minimum phase systems show phase responses which are much easier to control than their non-collocated counterpart.

modeled with a transfer function but the time delay cannot. The exact representation of the time delay is  $e^{-t_d s}$  and must be approximated by a rational function to be fully understood using system/control theory. It can be shown that an all-pass filter is a good approximation to a pure phase delay. Eq. 3.2 can then be rewritten as Eq. 3.3.

$$G(s) = G_{mp}(s)e^{-t_d s} aga{3.2}$$

$$G(s) \approx G_{mp}(s)G_{ap}(s) \tag{3.3}$$

Viewing the pole-zero map of the same system, with and without the system delay also provides useful insight as to why the time delay inherent to 3D acoustic systems causes such a significant degradation in controller performance. Fig. 3.2 identifies the



Figure 3.2 A 3D acoustic system model separated into its minimum phase part and its time delay (modeled using an all-pass filter approximation). As is often the case, more poles and zeros are needed to represent the time delay than the acoustic dynamics.

poles and zeros associated with the minimum phase portion of the response (shown in blue) and identifies the poles and zeros introduced into the system solely because of the time delay (shown in red). The right half plane (RHP) zeros from the all-pass filter approximation leads to a few control issues. For instance, the common control practice of plant inversion would not be possible since it would yield RHP poles, making the plant inverse unstable. Of course, controllers do not always require plant inversion.

The most important issue regarding RHP zeros is that increasing the gain of any controller far enough will lead to instability. This is because as controller gain increases, closed loop poles move from open loop poles to open loop zeros. When the open loop zeros are in the RHP, increasing the gain will move the closed loop poles into the RHP,



Figure 3.3 Root locus showing instability results with very little controller gain due to the RHP zeros pulling the poles into the RHP. The RHP zeros are a byproduct of the time delay.

causing instability. This concept is illustrated in Fig. 3.3, which shows the root locus of a similar acoustic system with RHP zeros. The curves show the movement of the closed loop poles as gain is increased. One can see that almost all of the poles become unstable with enough gain, and the few poles close to the  $j\omega$ -axis (the resonant modes) cross the  $j\omega$ -axis with very little increase in gain.

The time delay problem leaves the engineer with two possible solutions. One would be to design a low gain controller based on the plant model as it is. This approach would reduce noise minimally at resonant modes and increases noise minimally elsewhere. The other solution would be to reduce the controller bandwidth so much that the resulting model would not include any RHP zeros and would roughly approximate the plant within its narrow bandwidth. This approach would reduce noise significantly in the narrow bandwidth which has been modeled and increase noise slightly in the surrounding spectral region.

Reducing the bandwidth of the controller works since the number of RHP zeros is directly proportional to the bandwidth. The number of RHP zeros is also proportional to the distance between the speaker and the microphone. This distance should be reduced as much as possible. Time delays lead to RHP zeros, which in turn necessitate reduced bandwidth control. This is why in feedback control, 0D headset noise cancelation systems with collocated speaker/microphone setups can successfully reduce broadband noise but 3D noise cancelation systems with non-collocated speaker/microphone setups are fundamentally limited to tonal noise.

# 3.3 Non-minimum Phase System Decomposition

Decomposing a plant frequency response into its minimum phase response and nonminimum phase time delay is important for modeling an acoustic system exactly (i.e. without an all-pass approximation of the delay). Modeling just the minimum phase portion significantly reduces the order since all of the poles and zeros in the all-pass approximation can be described simply by  $e^{-t_d s}$ . Extracting the exact time delay is also critical if one wishes to counteract it within the controller. This so-called time advance filter is presented in the next chapter and is the central topic of this paper. But before we get there, we must be able to do the decomposition. There are several ways of going about it, each of them giving quite different results. Each method attempts to separate G(s) into  $G_{mp}(s)e^{-t_d s}$ .

1. Using the Bode gain-phase relationship definition of Eq. 3.4 should give an exact decomposition [17]. Unfortunately, with band-limited frequency response data it's

impossible to go to the infinite limits required in the equation. Integrating just over the relatively small bandwidth of the measured frequency response produces unsatisfactory results. In addition, experimentally obtained frequency response data is always somewhat noisy, so taking the derivative of the data amplifies the noise. The computation is also extremely slow.

$$\angle G_{mp}(j\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{dln |G(j\omega)|}{dln\omega} ln \left| \frac{\omega + \omega_0}{\omega - \omega_0} \right| \frac{1}{\omega} d\omega$$
(3.4)

2. The Bode gain-phase relationship from above can be simplified to Eq. 3.5 [17]. This calculation is much faster but it still has the same pitfall of taking the derivative of possibly noisy frequency response data. The simplification seems to effect resonances the most, making them more inaccurate. This is not good since modeling resonances are key to controller robustness and performance.

$$\angle G_{mp}(j\omega) \approx \frac{\pi}{2} \frac{dln|G(j\omega_0)|}{dln\omega_0}$$
(3.5)

3. Perhaps the simplest of all methods is simply to measure the physical distance between speaker and microphone and divide by the speed of sound to get the time delay as in Eq. 3.6. The drawback to this method is the difficulty in measuring both of these quantities. Measuring the speed of sound accurately requires an accurate measurement of temperature. Measuring the distance between speaker and microphone is problematic since neither is a point source. The speaker especially, as it is likely to be a large cone-shaped woofer. In addition, this method does not account for the electrical/sampling delay of the controller.

$$t_d = \frac{\text{distance}}{\text{speed of sound}} \tag{3.6}$$

4. The last method described uses the real cepstrum to decompose the measured frequency response [18]. This method proved to be very accurate. It also takes into account electrical/sampling delay since it uses the measured frequency response data as input. As seen in Eq. 3.7, it utilizes the FFT and is therefore extremely quick and can be implemented with just a few lines of code using Matlab. It is the preferred method of performing non-minimum phase decomposition.

$$G_{mn}(j\omega) = e^{F(\overline{x}.*F^{-1}(\ln|G(j\omega)|))}$$
(3.7)

where F is the FFT,  $F^{-1}$  the IFFT

and 
$$\overline{x}(k) = \begin{cases} 2 & \text{for } k = 2, 3, \dots, \frac{N}{2} \\ 1 & \text{for } k = 1, \frac{N}{2} + 1 \\ 0 & \text{for } k = \frac{N}{2} + 2, \frac{N}{2} + 3, \dots N \end{cases}$$

# 3.4 System Identification

System identification is the process of obtaining a mathematical model describing the input/output dynamics of a linear system. This mathematical model can be written in the form of a transfer function or as an equivalent state-space representation and can be obtained through either analytical derivation or curve-fitting of experimental data. This section focuses on the latter since obtaining experimental frequency response data of an acoustic system is much easier and more accurate than mathematically deriving the system model [19]. Obtaining an accurate system model is key to controller robustness

and performance.

3D acoustics problems with complex geometries are extremely difficult to solve analytically and their solution will most likely be a simplification of the actual system. On the other hand, obtaining a frequency response experimentally is not very difficult, a process made even simpler with the use of a dual-channel spectrum analyzer. Once the data has been collected, system identification programs must be used to fit a proper model to the data. In the case of acoustics and vibrations systems, more information about the system can be inferred from the frequency response than the time response, so frequency response system identification software is preferred.

The frequency domain system identification routine used in this paper is called SOCIT (System/Observer/Controller Identification Toolbox) and is courtesy of NASA. SOCIT makes use of the eigensystem realization algorithm to compute a discrete-time model from experimental magnitude and phase data using samples of the pulse response of the system (Markov parameters) [20]. This program is well-suited for acoustics problems because it outputs a model in state-space form which is well-conditioned and easily converts to modal canonical form. It is also able to identify very high order plant dynamics with little error in a very short time. Unfortunately, there are a few problems with the SOCIT algorithm, but these have been dealt with in previous work [21]. The resulting system identification procedure is a modification of the SOCIT toolbox which achieves extremely accurate system models.

#### 3.4.1 Obtaining the Frequency Response

Frequency responses of acoustic systems can be obtained quite easily. To get the necessary data, all one needs to do is play band-limited white noise through a speaker and record the resulting microphone data. Post processing the data using Eq. 3.8 will

yield the frequency response. There are several important parameters omitted from this equation which are left to the engineer's discretion, such as length of time data captured, number of averages for smoothing data, type of window used, and amount of overlap between successive FFTs.

$$G(\omega) = \frac{G_{yx}(\omega)}{G_{xx}(\omega)}$$
(3.8)

$$=\frac{Y(\omega)X^*(\omega)}{X(\omega)X^*(\omega)}\tag{3.9}$$

$$=\frac{FFT(y(t))conj(FFT(x(t)))}{FFT(x(t))conj(FFT(x(t)))}$$
(3.10)

It proves quite handy to be able to view the frequency response in real-time (as it is captured). For one, this enables the engineer to quickly identify problems which frequently occur such as a microphone amplifier not turned on or a filter not set to the proper cut-off frequency. Viewing the frequency response in real-time also lets one instantly see the effects of changing parameters in the system. This way, an undesirable pole or zero in the response can be avoided by changing the acoustics of the environment or by moving the speaker or microphone to a new location. Great intuition of the frequency response can be learned with the ability to instantly see the effects of changing the acoustic environment.

One way to see the frequency response in real-time is to obtain a dual-channel spectrum analyzer. This impressive piece of hardware has an output port for the speaker, an input port for the microphone signal, a screen which displays magnitude and phase, and a large menu of adjustable parameters. A spectrum analyzer is capable of much more than recording and displaying frequency response measurements, but for our application, this is the only needed capability. In an effort to provide a low cost alternative



Figure 3.4 Block diagram for viewing real-time acoustic frequency response data.

to the spectrum analyzer, an algorithm using Simulink was developed which computes frequency responses in real-time. A block diagram of this is shown in Fig. 3.4.

The algorithm shows how band-limited white noise is fed to both the speaker and the computer for processing using the STFT (Short-Time Fourier Transform). Once the noise has passed through the plant dynamics and been picked up by the microphone, it undergoes the same STFT processing. Within the STFT block, an appropriate buffer size, overlap amount, and window type must be chosen. In this project a Hann window with N/2 overlap was used (with N being the number of points in the FFT). It is advised to keep N as well as the sampling rate at a multiple of two so that the FFT computes at peak efficiency and the resulting center frequency bins will be clearly identifiable (i.e. 86 Hz rather than 85.7852...Hz).

Once the STFTs have been computed, it is easy to get the speaker/microphone crossspectrum and the speaker auto-spectrum. A key component of Fig. 3.4 is the averaging. Specifically, it must be done before the division to prevent the noise in the individual spectrums from being magnified. This will significantly prolong the time needed to achieve a smooth frequency response. Typically, in the frequency range of 0-1000 Hz with a center frequency bin every 1 Hz, no more than 10-20 seconds should be needed to arrive at an adequately smooth response. The acquisition time increases if more points are desired or a lower frequency range is needed.

From there, computing the magnitude and phase are pretty straightforward. One can decide whether or not to output the magnitude in dB. Also the choice is left to the user to output the phase, wrapped or unwrapped, in radians or degrees, as there are Simulink blocks which do this. At the tail end of the diagram are vector scopes which display the magnitude and phase of the frequency response. Provided the computations occur on a computer with zero latency input-output capability, these scopes will display accurate responses in real-time, giving the user great advantages over the post-processing method.

#### 3.4.2 Experimental Results

The experimental setup shown in Fig. 3.5 was made to mimic a large tractor cab enclosure. This particular acoustic environment is quite complex acoustically. Walls are about half covered in carpet and half plexiglass, providing a mixture of absorptive and reflective acoustics. The walls are also not at 90° and the ceiling has an irregularly shaped trapdoor. As can be seen in Fig. 3.6, the control speaker and microphone are spaced inside the cab approximately one meter apart. The disturbance speaker is located outside the cab. Below is a list of the necessary components to perform real-time frequency response measurements and real-time control.

 Speakers: 8" woofers were used for the control speaker and the disturbance speaker. It is important to get a speaker capable of playing noise in the desired control bandwidth (very low frequencies in most noise control applications). The speakers used have a frequency response from 30-3000 Hz. Be aware if a speaker has an in-built crossover (filter) as it will add its own dynamics to the overall plant.



Figure 3.5 Enclosure made to mimic a large tractor cab with the controller workstation in the foreground.



Figure 3.6 Interior of the enclosure showing the control speaker and microphone.

- 2. Speaker Amplifier: When choosing a speaker amplifier, it is very important not to get one with a digital filter or equalizer. This is a standard option in modern home theater receivers, so one needs to either look for old models (the one used here is over 20 years old) or look for simple power amplifiers which do not use digital processing.
- 3. Microphone: A low cost electret microphone will suffice for control applications although a nicer PCB microphone is shown in the figure. Both were used in this project. Sound level accuracy is not nearly as important as consistency of measurement.
- 4. Microphone Amplifier: The PCB microphone pictured comes with it's own battery powered microphone amplification unit. If using an electret microphone, a simple amplifier can be built according to the microphone's sensitivity. A circuit schematic is sometimes provided with the purchase of a microphone. Again, be aware of filters and what effect they will have on the frequency response.
- 5. Filters: Before microphone data is sampled by the processor, it's important to filter it with a low-pass filter with cutoff frequency somewhere below the Nyquist rate to prevent signal aliasing. Also, electret microphones need a DC bias to work properly, so before amplification, the signal needs to be high-pass filtered to eliminate the DC component. The high-pass cutoff frequency should be low enough to avoid interfering with the control bandwidth.
- 6. Processor: Active noise control requires a board with a minimum of 8-bit IO capability and 1 kHz sampling rate with very little to no latency. Having as many inputs and outputs as needed is of course important. Many DSP boards come with

lots of inputs and few to no outputs, so the choice is limited when looking for a board with multiple outputs in addition to inputs. At the extra cost of increasing the IO count, resolution, or sampling rate, one gets more diverse functionality with potentially increased performance. The 1103 DSpace controller board and the Measurement Computing PCI-DAS1602-16 processor were both used for this project. A PC with audio card was attempted for use as a controller, but was found to have too much latency for real-time applications due to the overhead of the operating system. It is therefore recommended to use a stand-alone DSP board.

7. Software: Using Matlab and Simulink has been a huge time-saver in algorithm development. If using Simulink, make sure to get a DSP board that is compatible. That way, a block diagram created in Simulink can be automatically converted into code and uploaded to the board. Programming the board directly in C is more computationally efficient but takes a lot more time of the engineer.

All of this equipment combined with the algorithm shown in Fig. 3.4 was used to view the frequency response of the cab enclosure in real-time. As the microphone is moved, the magnitude and phase response subtlety shifts, resonant peaks and valleys shifting in frequency and getting more or less damped. The response shown in Fig. 3.7 came from the speaker and microphone orientation in Fig. 3.6. This response shows a large phase delay which can be attributed to the one meter separation distance between speaker and microphone. As explained early in this chapter, a response with this large amount of phase delay would certainly be modeled with many RHP zeros, designating it non-minimum phase and making it very difficult to control.



Figure 3.7 The frequency response of the tractor cab is extremely complex with large phase lag.

Using the real cepstrum technique for decomposing a non-minimum phase system, we can input the frequency response data,  $G(j\omega)$ , from Fig. 3.7 into Eq. 3.7. The output is the minimum phase response,  $G_{mp}(j\omega)$ , which is plotted in Fig. 3.8. Also plotted in Fig. 3.8 is the system model identified using the modified SOCIT software described in the previous section. As can be seen from the near identical curves, this system identification software performs extremely well. Within the controller bandwidth of 30-800 Hz, the 80 state model averages just 0.217 dB magnitude error and 1.79 ° phase error.

The pole-zero map of the system is plotted in Fig. 3.9. It has 80 LHP poles and 80 LHP zeros, so it is stably invertible which will be a necessary condition for the model reference control implemented later. Since the non-minimum phase decomposition took the large phase lag out of the frequency response, the model of the resulting response



Figure 3.8 The 80 state model of the enclosure frequency response has an average magnitude error of only 0.217 dB and an average phase error of only  $1.79^{\circ}$  from 30-800 Hz.



Figure 3.9 The model of the minimum-phase acoustic system shows no right half plane zeros and has an equal number of poles and zeros so it is stably invertible.

contains no RHP zeros. This minimum phase plant would be much easier to control if the time delay were to disappear. Of course the physical time delay can never disappear, but the next chapter will attempt to minimize its effects by introducing a so-called time advance filter in the controller.

## 3.5 Remarks

The 3D active noise control time delay problem was shown to be the cause of performance and bandwidth limitations because it introduced RHP zeros into the plant dynamics. Non-minimum phase decomposition was introduced as a means of circumventing this problem and was used as a tool for singling out the problematic time delay. The real cepstrum method proved to be the best decomposition technique. A method for dealing with the identified time delay is left to the next chapter. A frequency response was taken of an example 3D acoustic enclosure which was successfully decomposed and accurately modeled with no RHP zeros.

# CHAPTER 4. TIME ADVANCING WITH THE STFT

# 4.1 Introduction

In this chapter, a so-called time advance filter is introduced which uses the STFT to shift the phase of a signal's multiple frequency components the precise amount needed to appear time advanced. It is the hope that using this filter will negate the ill effects of the time delay which greatly restrict controller efficacy. The procedure used to deconstruct a signal into small segments using the STFT and then reconstruct it back using the ISTFT is given. This is needed since the time advance filter acts in the frequency domain. Several simulation results are given which show the performance and limitations of the time advance filter.

## 4.2 Controller as Time Advance Filter

A non-causal system is one in which output relies on future input [22]. Therefore any physical system must be causal to satisfy the laws of nature. It is impossible to break the fundamental causality constraint of all physical systems and advance a signal in time. There are however a few ways to achieve time advancement without breaking causality. One which is often used obtains an appropriate length signal preview. This method is the basis for feedforward control and is made possible since the electric time delay resulting from the controller filter is less than the acoustic time delay between the preview microphone and the controller speaker. With prior knowledge of the disturbance and the acoustic path, a feedforward controller is able to reduce noise at a separate performance (or error) microphone. The major assumption that feedforward control makes is for the disturbance at the preview microphone to be well correlated with the disturbance at the performance microphone. If not, controller performance will suffer and noise reduction will be small.

The new method introduced in this paper does not use a preview of the incoming acoustics as in feedforward control. Instead, this time advancement is accomplished with feedback control, using only a performance microphone. It is able to achieve a time advancement without breaking causality by relying on the periodicity of the disturbance acoustics. The essential idea is that if one assumes a signal is periodic, even for some short duration of time, one can predict future values of the signal by looking at current values. Even though the output of the controller will occur after the input is read, the signal will appear to be time advanced. This is because when a sine wave is delayed by  $360^{\circ} - \theta^{\circ}$  it is identical to advancing the same wave by  $\theta^{\circ}$ .

A time delay is represented in the Laplace domain as  $e^{-t_ds}$ , and so it follows that a time advance is represented as  $e^{t_ds}$ . Approximation of this  $e^{t_ds}$  filter is our goal. It is apparent why when inspecting the closed loop block diagram of Fig. 4.1 and the resulting closed loop transfer function in Eq. 4.1. As the previous chapter indicated, acoustic plant dynamics can be decomposed into a minimum phase part and a time delay. If one can then construct a time advance filter in the control loop based on the time delay information provided by the decomposition, one can effectively cancel out the effect of the time delay. This would be very beneficial since the acoustic delay causes large problems for control. The resulting closed loop transfer function of Eq. 4.1 would appear to behave exactly like a plant with collocated speaker and microphone.



Figure 4.1 Block diagram of closed loop feedback system showing decomposed plant, controller H(s), and time advance filter  $e^{t_d s}$ .

Controller performance of such collocated systems significantly outperform others.

$$\frac{E(s)}{D(s)} = \frac{1}{1 + G_{mp}(s)e^{-t_d s}H(s)e^{t_d s}}$$
$$= \frac{1}{1 + G_{mp}(s)H(s)}$$
(4.1)

# 4.3 Signal Deconstruction and Reconstruction

The Short Time Fourier Transform (STFT) means simply taking an FFT of a small segment of signal. Breaking up a signal into multiple time segments and taking the STFT of each allows one to see how the frequency information of a signal changes with time. Putting these STFT's together into a colormap picture (called a spectrogram), one is able to see how a signal's spectrum changes with time. Spectrograms provide a very nice visual representation of spectrally varying sound and so will be used to show all of the controller performance results at the end of this paper. If a signal is deconstructed into



Figure 4.2 Illustration of the overlap-add reconstruction of a signal which has been deconstructed into several windowed segments.

the frequency domain using the STFT, it is reconstructed in the time domain using the inverse STFT (or ISTFT). It is using this process of STFT deconstruction and ISTFT reconstruction that we will implement the time advance filter.

STFT parameters include sampling rate, buffer size, hop size, and window type. A few of these parameters are illustrated in Fig. 4.2. A generic signal's corresponding deconstruction and reconstruction are shown as well. The sampling rate  $(f_s)$  is the rate at which the continuous signal is sampled by the processor's A/D converter to make the discrete signal. The buffer size (N) is the length of time of the STFT segment, given as an integer number of samples of the discrete signal. The hop size is the number of samples between one STFT segment and the next. The smaller the hop, the larger the overlap between successive STFTs.

Windowing is needed to prevent the spectral distortions that arise when the beginning and end of the time segment do not smoothly match up. These distortions are reduced by multiplying a (generally bell-shaped) window to the time segment so that the ends more smoothly match up. For perfect signal reconstruction, these windowed segments must overlap and add to a constant. This is aptly named the Constant Overlap-Add (COLA) condition and is shown mathematically in Eq. 4.2 and graphically in Fig. 4.3 [23].

$$x = \sum_{m} x_{m} \quad \text{iff}$$
$$\sum_{m \in \mathbb{Z}} w(n - mh) = c \quad \forall n \in \mathbb{Z}$$
(4.2)

where m is the window index, h is the hop size, and c is a constant value.

Eq. 4.2 says that perfect reconstruction will occur if and only if all of the windowed segments (w), shifted by some hop size (h), add up to a constant (c). This is perhaps better shown than explained. In Fig. 4.3, multiple Blackman-Harris windows (red) are summed together to form the combined reconstructed window (blue). This overall window must be a constant (after some small startup time) for the COLA condition to hold. The plot is shown with three different hop sizes to show the effect of hop size on the reconstructed signal. From the third plot one can see that the COLA condition will fail with too large a hop. In fact, there is a maximum hop size required to meet the COLA condition for each window.

The table in Fig. 4.4 shows all of the window functions available in Matlab. Four of the windows never satisfy the COLA condition regardless of hop size. Of the ones that do, their maximum hop size needed for perfect reconstruction is identified. A window with a large maximum hop size would be desirable if computational cost is paramount. This is because smaller hops result in more windows, and more windows result in more STFTs and ISTFTs that need to be computed in the same length of time. One can



Figure 4.3 Progression illustrating a window's maximum hop size parameter as well as its role in signal scaling.

Window	COLA	<u>Max Hop</u>	Scale
Bartlett	Yes	N/2	1
Bartlett-Hanning	Yes	N/2	1
Blackman	Yes	N/3	1.26
Blackman-Harris	Yes	N/4	1.435
Bohman	No		
Chebyshev	No		
Flattop	Yes	N/5	1.07789475
Gauss	No		
Hamming	Yes	N/2	1.08
Hann	Yes	N/2	1
Kaiser	No		
Nuttall	Yes	N/4	1.4543276
Parzen	Yes	N/4	1.5025
Rectangular	Yes	Ν	1
Tukey	Yes	N/4	3

Figure 4.4 Comparison of window parameters important to signal reconstruction.

also see from Fig. 4.3 that the smaller the hop, the larger the reconstructed signal will be. Therefore, the signal must be scaled by some factor to get perfect reconstruction. This scaling factor is also given in the table. Eq. 4.3 shows how the scaling factor, hop size, and maximum hop size for a given window are all used in the scaling of the windowed segments to achieve perfect reconstruction of the original signal. A comparison of windows as a function of their accuracy of reconstruction of various signals will be provided in the next chapter.

$$x = \frac{1}{\text{scale}} \left(\frac{\text{hop}}{\max \text{ hop}}\right) \sum_{m} x_m \tag{4.3}$$



Figure 4.5 Multiplication with  $e^{j\omega t_d}$  linearly increases phase without changing magnitude.

# 4.4 STFT Time Advancement

In between the deconstruction and reconstruction of a signal is where the frequency domain signal manipulation takes place. A time advance in the time domain is achieved by shifting the signal forward in time. A time advance in the frequency domain is achieved by multiplication of the Fourier transformed signal by  $e^{j\omega t_d}$ . This is shown in Eq. 4.4 and is simply the time shift property of the Fourier transform [22]. A plot of the frequency response of  $e^{j\omega t_d}$  is given in Fig. 4.5. The magnitude of the filter is unity for all frequencies and the phase increases linearly with frequency and has a slope of  $t_d$ .

$$x(t+t_d) \Leftrightarrow X(\omega)e^{j\omega t_d}$$
 (4.4)


Figure 4.6 STFT time advancement block diagram with example signal to illustrate each step.

Using the STFT, we are able to identify magnitudes and phases of discrete frequency components of individual signal segments. When multiplying by  $e^{j\omega t_d}$ , the magnitude is unchanged but the phase is increased linearly with frequency. This will phase shift each frequency of the signal the appropriate amount to appear time advanced. A block diagram showing the STFT time advancement procedure is given in Fig. 4.6. This is the inner workings of the  $e^{t_d s}$  block from Fig. 4.1.

In Fig. 4.6 the buffer acts to gather N number of samples of the time signal. The buffered signal is then windowed by a user defined N length window which satisfies the COLA condition. An FFT then transforms the signal into the frequency domain. The frequency domain signal is then multiplied by  $e^{j\omega(t_d+N/f_s)}$ . Notice that this includes an added time advancement of  $N/f_s$ . This is because in order to gather N samples of data into the buffer, a time lag of  $N/f_s$  is incurred. By increasing the slope of the time advance filter's phase by  $N/f_s$ , we negate the effect of this time lag. An IFFT is then



Figure 4.7 The STFT filter perfectly reconstructs a time advanced 32 Hz sine wave.

performed on the output to convert the signal back into the time domain. This process is completed at every hop interval on each N point overlapping signal segment.

## 4.5 STFT Filter Performance

To test the performance of the time advance filter, several sine waves are used as inputs and the resulting outputs are observed. The filter is judged by how well the time advanced output matches up with the input at a future time. If the sine waves perform as expected, more complex signals will be attempted.

For the first test, a 32 Hz sine wave is used as input to the filter. The results are plotted in Fig. 4.7. It is assumed the speaker and microphone have an acoustic time delay of  $t_d$  seconds which separate them. When the disturbance sinusoid is played, the



Figure 4.8 The STFT filter incorrectly reconstructs a time advanced 34 Hz sine wave.

signal is first captured by the microphone and buffered for the initial  $N/f_s$  seconds. Once the data is acquired, the processor can compute the time advanced signal using the steps outlined in the previous section and then output it at  $t = N/f_s$ . Once it leaves the speaker, the noise takes  $t_d$  seconds to arrive at the persons ear (or performance microphone). As can be seen from Fig. 4.7, the 32 Hz sinusoid is accurately time advanced since the input and output match. For control purposes, the output would be negated before being played through the speaker so that the waves would cancel each other. This is not shown because it would be harder to tell how well the signals align.

For the next test, a 34 Hz sine wave is used as input to the filter. The results are plotted in Fig. 4.8. This time the input and output do not match up. In fact, no frequencies match up except ones lying exactly at center frequency bins of the FFT. To understand the reason for this, one needs to understand the limitations imposed by the FFT. The FFT takes in a discrete number of time data points, N, and transforms them into N frequency data points. Therefore, a perfectly detailed frequency response can never be known and it is always approximated by a discrete number of points. Eq. 4.5 shows the relationship between a standard time vector and its corresponding frequency vector once the FFT is taken. The first number in the vector is the starting point, the last number is the end point, and the middle number is the spacing between values.

From Eq. 4.5, we see that the equally spaced N point time vector is translated into an equally spaced N point frequency vector. However, half of the frequency vector is filled with negative frequency bins. These are just the complex conjugate of the positive frequency bins and are therefore redundant. The amount of useful frequency data is then cut in half, where the useful frequency vector is  $[0: f_s/N: f_s/2]$ . This means that taking the FFT of an N point time signal effectively yields N/2+1 frequency data points. This makes sense when one remembers the Nyquist rate. From the Nyquist sampling theorem, it is shown that sampling at  $f_s$  gives a maximum observable frequency of  $f_s/2$ . This means the largest resolvable frequency still requires two time points as there are time data points.

When the 34 Hz sine wave was input into the STFT time advancing filter it was discretized into frequency bins of [0 16 32 48 64...] and thus was identified as its nearest

neighbor, 32 Hz. The 34 Hz signal was then given the phase advance associated with a 32 Hz signal which led to the inaccuracy of the output. One might think a possible solution to this problem would be to simply increase the number of points in the FFT, N, thereby reducing the center frequency bin separation,  $f_s/N$ . One problem with this is that increasing N means that more data needs to be buffered, increasing the time lag between when the microphone reads the data and when the speaker can play a canceling wave. This lag needs to be as short as possible so that the time advancing filter can work for real world signals which are periodic with the period slowly changing with time.

Increasing N does increase the number of frequencies available for accurate time advancement, but it will always be a finite number of frequencies in a world capable of producing infinite frequencies. This point is illustrated in Fig. 4.9. The figure shows the error between a signal and its STFT reconstructed signal. The x-axis shows the length of data taken, which doubles each time. The y-axis is zoomed in to a small frequency range to properly see the fractal effect. The figure shows error is small at the discrete center frequency bins, but only the discrete frequency bins. All frequencies in between exhibit large error. Increasing the length of data sampled does nothing to effect the likelihood of a random frequency having large error when time advanced.

The result of Fig. 4.9 shows the STFT time advancement cannot work unless the disturbance frequencies happen to match the center frequency bins of the FFT. For example, if a one second buffer is used to capture a 64 Hz sine wave, the STFT time advancement will identify the 64 Hz signal exactly and advance the phase the appropriate amount since 64 Hz is a center frequency bin. However, if in the future the signal changes slightly to become 64.5 Hz, the STFT reconstruction will be completely out of phase since 64.5 Hz is in between two center frequency bins. The phase advancement for the 64.5 Hz wave will have to be approximated to that corresponding to 64 or 65 Hz. This



Figure 4.9 Increasing the buffer time increases the number of accurately time advanced frequencies. No matter how large the buffer, though, there will always be large error in the spaces between center frequency bins of the FFT.

may not seem like a significant error, but since the buffer is one second, by the end of that second a 0.5 Hz error makes the signal significantly out of phase.

# 4.6 Remarks

The idea of a time advance filter was proposed as a way of combating the time delay problems inherent to 3D noise control. In theory, the successful implementation of such a filter would offer the same excellent closed loop performance as a collocated noise controller. The procedure for how to implement such a filter in the frequency domain was presented with detailed information about deconstructing and accurately reconstructing a signal. The STFT time advancement filter was shown to produce accurate time advanced signals only for a finite number of frequencies corresponding to the FFT center frequency bins. Since this would fail with most real world disturbances, more work needs to be done to make time advancement accurate for all frequencies. This work is done in the next chapter.

# CHAPTER 5. TIME ADVANCING WITH THE PHASE VOCODER

## 5.1 Introduction

In this chapter, a phase vocoder will be used to provide quick frequency and phase estimates of an incoming signal. This will greatly improve functionality of the time advance filter presented in the previous chapter. Specifically, it will allow for accurate time advancing not just at FFT frequency bins but at all frequencies up to the Nyquist rate. Several test signals will be used to evaluate the performance of the filter. Sources of error will also be presented and quantified to help the reader understand the limitations of the algorithm. The impact on performance of various parameters such as window size and type will also be discussed.

## 5.2 The Phase Vocoder

The phase vocoder is primarily used as a tool to slow down or speed up an audio signal without altering its pitch. It can achieve this effect by manipulating the signal in the frequency domain using the STFT. Although the end result of time scaling is very different than time advancement, the initial steps of the phase vocoder can be implemented to serve our purposes. What separates the phase vocoder from ordinary STFT signal deconstruction and reconstruction is that it takes past phase information into account when determining the frequency content of the signal. Using this past phase information gives the phase vocoder much greater accuracy in its frequency estimation than the STFT, which completely ignores this phase information.

#### 5.2.1 PV Frequency Estimation

Frequency estimation with the STFT is rather limited. For example, if we wanted to estimate the frequency of a single tone, the STFT could only show precision to the nearest center frequency bin. On the other hand, using the phase vocoder, we can achieve much greater precision simply by comparing the phase of the current STFT with the phase of the previous STFT. With the knowledge of how much the phase of a signal has advanced over a set time, the frequency of the signal can be determined with a high degree of accuracy. Eq. 5.1 shows how one can find the frequency given two phases separated in time [24].

$$f_n = \frac{(\theta_2 - \theta_1) + 2\pi n}{2\pi \Delta t}$$
(5.1)
where  $\Delta t = \frac{hop}{f_s}$ 

#### and n = number of encirclements.

This equation is quite simple but very powerful.  $\theta_1$  and  $\theta_2$  can be single values centered around a single frequency bin or they can be vectors of values obtained from the STFT, with the phases corresponding to the frequency vector  $[0: \frac{f_s}{N}: \frac{f_s}{2}]$ .  $\Delta t$  is the length of time that has elapsed between successive STFTs. The role that *n* plays can be more fully understood by looking at Fig. 5.1 [24].

Let's say the phase of the first segment is identified as  $\theta_1$  and the phase of the second



Figure 5.1 Comparing successive phases yields a different frequency estimate for each n (see Eq. 5.1).

segment is is identified as  $\theta_2$ . Depending on the length of  $\Delta t$ , the signal's wave could still be in the current cycle (n = 0), it could have traveled a full revolution (n = 1), two revolutions (n = 2), or more. Since phase values are cyclical and can only take values from 0 to  $2\pi$ , the number of cycles occurring between segments cannot be known exactly. However, since it is known how many cycles have elapsed at each STFT center frequency bin, it can be assumed that the same number of cycles have elapsed for the frequency identified by the phase vocoder at that bin. A modification of Eq. 5.1 can then be written which does not contain the arbitrary variable n. Eq. 5.2 shows this modification using vector notation.

$$\vec{f} = \frac{\left(\vec{\theta}_2 + \left[0 : 2\pi \left(\frac{hop}{N}\right) : \pi(hop)\right]\right) - \vec{\theta_1}}{2\pi\Delta t} \tag{5.2}$$

Assuming  $\vec{\theta_1}$  and  $\vec{\theta_2}$  are phase vectors computed from the STFT of two consecutive signal segments, we can add the expected phase advance of  $[0:2\pi(\frac{hop}{N}):\pi(hop)]$  to  $\vec{\theta_2}$  to

get the actual, unwrapped phase of  $\vec{\theta_2}$ . By computing the difference between successive phases divided by the time lapse between STFT segments, we can arrive at our goal of finding a frequency vector which accurately matches the frequency components of the signal.

#### 5.2.2 Performance

In an effort to see how well the phase vocoder does with estimating frequencies, a few simulations have been run. The first is shown in Fig. 5.2. Here, a chirp signal is given as the test signal and runs from 100 Hz to 400 Hz over the course of 10 seconds. This linearly increasing frequency will be continually estimated using both the STFT and the phase vocoder. The parameters for both are: 2048 Hz sampling rate, 32 sample buffer, 16 sample hop, using a Hann window.

The results of Fig. 5.2 show the phase vocoder significantly outperforms the STFT. The STFT is only capable of estimating the signal frequency at center frequency bins which are very largely spaced using these parameters. This is the reason the estimates only fall on discrete points 64 Hz apart. On the other hand, the phase vocoder shows near perfect accuracy in estimating the chirp signal. The average error over the course of the 10 seconds for the phase vocoder is just 0.1 Hz while the STFT has an average error of 16 Hz. Also of note, if a slower chirp is used, the phase vocoder shows improved accuracy while the STFT accuracy remains the same.

The second simulation shows how quickly the two methods are able to achieve accurate frequency estimation. To generate Fig. 5.3, a 100 Hz sine wave was used as the test signal while the buffer length was varied. The figure shows that the frequency estimation error using both methods fluctuate significantly, but at the same time both follow clear trends. For the STFT curve, the points of least error correspond to when



Figure 5.2 With a short preview window, STFT frequency estimation is coarse while the phase vocoder is able to follow the chirp to within 0.1 Hz.



Figure 5.3 Frequency estimation vs preview time (window size) using the STFT and PV methods.

the time matches with the end of a cycle of the sine wave. The accuracy is relatively poor in between these points even though a Hann window was used. Despite these fluctuations, it is clear that the phase vocoder is significantly better than the STFT at frequency estimation. To achieve accuracy comparable to the phase vocoder, the STFT would require a very long buffer time. This long length of time is impractical for many applications, including active noise control.

There are a few important parameters that can be changed which effect the figure. For one, increasing the test signal frequency doesn't change the STFT error. However, using the phase vocoder, a decrease in error by a factor of four is observed per doubling of frequency. This is quite significant and essentially means that the phase vocoder's frequency detection accuracy is very dependent on how many signal wavelengths are captured in the buffer. Through experimentation, the upper limit of acceptable frequency estimation error (< 1 Hz) is generally reached with a buffer capturing just 1-2 wavelengths of signal. Lowering the hop size parameter can reduce this minimum buffer size by improving the error for short buffer sizes. Unfortunately, this also has the effect of increasing the error of frequency estimation when using long buffer sizes.

## 5.3 PV Time Advance Filter

In the last chapter we were able to create a time advance filter using the STFT. However, it was impractical since it could only get an accurate measurement of the phase at discrete frequency bins. The phase vocoder has been shown capable of identifying a continuous range of frequencies with excellent accuracy. This ability allows for the accurate measurement of phase needed for time advancement. We will therefore augment the STFT time advance filter with a phase vocoder to extend its functionality to all frequencies.

Using a phase vocoder for time advancement is very similar to using it for frequency estimation. The first step to estimating a frequency is determining the phase change that occurs between successive STFTs. The phase vocoder's very accurate measurement of change in phase is all that is needed for time advancement. This is because a signal's future phase advance can be estimated with knowledge of the signal's previous phase advance. This is the core principle of this chapter and is more explicitly stated in Eq. 5.3 and Eq. 5.4.

$$\phi_{ADV}(\omega) = \phi_t(\omega) + \phi_{t_d}(\omega) + \phi_{N+hop}(\omega)$$
(5.3)

where  $\phi_{ADV}(\omega)$  is the time advanced phase,

 $\phi_t(\omega)$  is the current phase,

 $\phi_{t_d}(\omega)$  is the acoustic phase advance,

and  $\phi_{N+hop}(\omega)$  is the electronic phase advance

$$\phi_{ADV}(\omega) = \phi_t(\omega) + (\phi_t(\omega) - \phi_{t-t_d}(\omega)) + (\phi_t(\omega) - \phi_{t-hop}(\omega)) \left(\frac{N+hop}{hop}\right)$$
(5.4)

From Eq. 5.3, we see that the proper phase advance is obtained by adding the expected acoustic phase advance and the electronic phase advance to the current phase. In the time advance filter, these phases are vectors derived from each STFT. The expected acoustic delay,  $t_d$ , is the time it takes for the control signal to get from the speaker to the microphone at the user's ear. This parameter is variable and is found using the ultrasonic tracking system described in chapter 2. The electronic phase delay is dependent on the buffer and hop sizes. The electronic delay required by the phase vocoder is N + hop samples since it needs to compare the phase information of two overlapping N-point STFTs separated by hop samples.

To obtain the most accurate phase advance, the acoustic and electronic phase advances should be calculated with their own individual phase vocoders. These phase vectors are obtained from subtracting phase of the delayed signal from the phase of the current, undelayed signal (shown in Eq. 5.4). In the case of the electronic phase advance, a delay of only *hop* is required to compute the phase advance since the overall delay of N + hop is a multiple of *hop*. When scaled by (N + hop)/hop, the appropriate electronic phase advance is acquired.

A block diagram of the time advance filter using the phase vocoder is given in Fig.



Figure 5.4 Block diagram of time advance filter using phase vocoder.

5.4. This is essentially Eq. 5.4 implemented as a real-time algorithm. The STFT and ISTFT blocks are used to simplify the drawing since they include many operations such as buffers, windows, and overlapping/adding of STFTs. This new algorithm will attempt to accurately time advance a periodic noise disturbance. If successful, it will be the primary mechanism of the active noise controller.

#### 5.3.1 Performance

The first test of this new time advance filter will be the same test the STFT filter failed in the previous chapter. The test signal was a 34 Hz sine wave. The STFT filter proved incapable of advancing the signal accurately since it did not exactly coincide with one of the FFT bin center frequencies. Fig. 5.5 shows the performance of the time advance filter augmented with the phase vocoder. As can be seen, the test signal and its time advanced counterpart match up nearly perfectly. What follows is a detailed description of the inner workings of this system (refer to Fig. 5.4).

First, the microphone captures the first N samples of the disturbance signal and



Figure 5.5 Time advance filter with the phase vocoder accurately reproduces previously inaccurate 34 Hz tone.

performs the FFT to get the phase vector needed. After hop samples pass, another N point FFT is taken to get the next phase vector needed. The first phase vector is considered  $\phi_{t-hop}(\omega)$  and the second phase vector  $\phi_t(\omega)$ . The electronic phase advance of hop samples is obtained by subtracting these two phase vectors. The total electronic phase advance of N + hop samples is arrived at by multiplying this by (N + hop)/hop. To find the acoustic phase advance, the incoming signal is delayed by  $t_d$  and the FFT performed to find the phase vector. This phase vector is then subtracted by the current (second) phase vector to give the acoustic phase advance.

Finally, the total phase advance is obtained from adding these two vectors (electronic and acoustic) to the current phase vector. This phase vector is then combined with the magnitude vector from the most current FFT and the IFFT is performed to transform the signal back into the time domain. This time advanced signal is played from a speaker at time  $(N + hop)/f_s$  and will arrive at the microphone (ear of user)  $t_d$  seconds later. Since the disturbance noise matches so well with this time advanced output, they would cancel each other if the output were negated before being played by the speaker.

In the next performance test, the output will be negated and added to the input so that it simulates feedforward control and the resulting error can be viewed. The input this time will be a ten tone signal with each tone spread out from 50 to 972 Hz. This multi-tone signal emphasizes an advantage of using FFT processing since the algorithm does the same amount of computations regardless of the complexity of the disturbance signal. This is because the whole filter uses vector operations and therefore does not need to continually adapt to a changing disturbance.

From Fig. 5.6, it is clear that the filter performs well since the error is over 30 dB less than the input signal strength and would certainly have potential as a feedforward active noise controller. From 0 to 0.094 seconds, the time advance filter takes input but does not produce any output. It finally outputs a signal based on the processing of the first 0.094 seconds, but the signal must travel through the air for 0.016 seconds until it can start canceling the disturbance at 0.11 seconds. The portion of the signal from 0.11 to 0.14 seconds is where the reconstructed output is first ramping up to a constant value from the add-overlap of windows (see previous chapter). After this portion, the reconstructed output windows add up to a constant value and so can cancel the input disturbance at its full potential.

Fig. 5.7 displays the error from the same multi-tone test in the frequency domain. From the magnitude plot we can see that the peaks of the input and output tones match well but do not match in between tones where there is little to no signal. One important thing to note is that there is more error in the regions where the tones are more closely spaced. This limitation will be analyzed further in the next section. Looking at the



Figure 5.6 Multi-tone PV filter error in time domain. Signal is analyzed in first portion and canceled in second.

phase error plot, it is evident that the phase is also matched well at the peaks with virtually no error, but elsewhere the phase does not match at all. This makes sense and has no ill effects since the signal is not present there.

The key advancement from the phase vocoder is the ability to time advance signals of any frequency, and as just demonstrated, signals having multiple frequencies. One unfortunate drawback is the phase vocoder algorithm requires  $(N + hop)/f_s$  seconds of signal before it can output, whereas the STFT algorithm required just  $N/f_s$ . This increase of input/output delay can be shortened by decreasing the hop at the cost of more controller computations. Either way, the needed time delay is quite short since, as shown earlier, the accuracy is quite good even when the delay covers just 1-2 wavelengths of the signal.



Figure 5.7 Multi-tone PV filter error in frequency domain shows accuracy at tones but no where else. Magnitude plot also shows input and output of filter.

#### 5.3.2 Error

The time advance filter using the phase vocoder appears to perform quite well under certain controlled test cases. However, there are several types of signals which the filter does not handle as well. There are three known causes for error in the time advancement: random noise, small tonal separation, and time varying periodic noise such as a chirp signal. These three limitations will be described in further detail in this section.

The inability of the time advance filter to operate correctly with random noise is explained by the early assumption made for the existence of a time advance filter. As stated in the previous chapter, the causality constraint would be violated if it were possible to predict future values of a random signal. The only way to not break the law of causality is to either obtain a preview of the signal (feedforward control) or assume periodicity in the signal. Since we are attempting to use feedback control, we must require a signal be periodic for proper time advancement. However, this does not mean that if a signal contains both periodic and random noise components that it will not be at all functional. In this case, the periodic components will be time advanced and the random components will not.

As hinted in the previous section, a signal's tonal separation can lead to time advancing errors. This limitation again stems from the FFTs finite nature. Since the FFT frequency bins are finite and rather coarsely spaced with such a short time window it becomes impossible to differentiate between two or more tones as they move closer to each other in the frequency domain. In an attempt to quantify this error, Fig. 5.8 shows the time advancement error versus tonal separation. To generate this plot, a dual-tone signal with one tone at 100 Hz and one at 1100 Hz was sent through the filter and the error of the output was calculated. The high frequency tone was then stepped closer



Figure 5.8 PV time advance filter error versus tonal separation for several window sizes.

and closer to the low frequency tone, with the error of time advancement calculated at each step.

Four different window sizes were used since that parameter has a direct effect on frequency resolution and therefore error. If the window size is 1/16 second, that means the corresponding FFT bin resolution is 16 Hz. A window size of 1/2 second gives 2 Hz resolution, etc. Clearly the error can be significantly reduced by increasing the window size. In fact there is approximately 18 dB less error per doubling of window size. Despite this, a huge window size is not advisable as it would hinder the filter's ability to adapt to a changing disturbance. Therefore, before choosing a window size one must determine how much the disturbance noise changes with time, or in other words, how transient the signal appears.

To help aid in the choice of proper window size as well as quantify another key



Figure 5.9 PV time advance filter error versus chirp speed for several window sizes.

source of error, Fig. 5.9 plots the time advancement error versus chirp speed. A chirp is a linearly varying frequency signal, and as such, provides a good model of a transient signal. Increasing the speed of the chirp in effect increases the transient behavior of the signal. To generate Fig. 5.9, a chirp signal starting at 200 Hz and increasing just 0.01 Hz per second for 10 seconds was used. Then the speed at which the chirp moved was increased incrementally until it reached the fastest speed of 100 Hz per second. As seen from the figure, a quicker chirp results in a less accurate reconstruction. Unfortunately, the error was quite large even for a slowly changing signal. This can be improved somewhat by decreasing the hop size, but at the cost of more computations.

The same four window sizes shown in Fig. 5.8 were used in Fig. 5.9 so that a proper comparison could be made. With regards to tone spacing, there was an 18 dB decrease in error with each doubling of window size. In the case of chirps/transients, there is

approximately 12 dB *more* error per doubling of window size. This fact highlights the crucial tradeoff involved in FFT-based control. It is a tradeoff of the window size parameter which can only be resolved with knowledge of disturbance signal characteristics. For instance, if the disturbance signal was known to have many closely spaced, slowly changing tones, a large window size would be chosen. Likewise, if the disturbance signal was known to have a few greatly spaced, quickly changing tones, a small window size would be chosen. If the disturbance signal was known to have several closely spaced, rapidly changing tones, a compromise window size could be chosen but the performance of the filter would likely be bad since it would be incurring error in two ways.

Besides the window size and hop size, another parameter which influences the error is the window type. For the tests above, the Hann window was used. However, there are ten other windows supporting perfect reconstruction (COLA compatible windows) which were identified in the previous chapter. Error curves for each of these windows are generated in Fig. 5.10 and Fig. 5.11. To show all 11 error curves on the same figure without getting too cluttered, the plot is shown as a colormap image with the color indicating the amount of error. For both plots, the window size was set at 1/4 second and the hop size at 1/32 second.

Fig. 5.10 shows the time advance filter error versus tonal separation for each of the 11 windows. From the plot is not obvious which type of window is best, but it seems the Blackman and Hann windows have the lowest error for a wide range of tonal separation. The Hann window was chosen over the Blackman window because it had slightly less error at 5 Hz tonal separation. The case might be made for choosing the rectangular window if it was known the tonal separation of the disturbance was extremely small. The risks would be high, though, since the error is quite bad for most tonal separations besides a few closely spaced ones at 5 and 9 Hz.



Figure 5.10 PV time advance filter error versus tonal separation for each window type.



Figure 5.11 PV time advance filter error versus chirp speed for each window

type.

84

Fig. 5.11 shows the time advance filter error versus chirp speed for each of the 11 windows. From the plot it seems that window type does not have nearly the same effect for chirp speed error as it did for tonal separation error. For slow to moderate chirp speeds, there is no difference between any of the windows except the rectangular window for which there is a clear disadvantage. For chirp speeds higher than about 6 Hz/sec the error is more than 0 dB, meaning the time advance filter is ineffectual for fast transient disturbance signals. For better handling of chirp signals, an extension of the phase vocoder could be implemented which linearly extrapolates the changes in phase to predict future frequencies.

## 5.4 Remarks

The phase vocoder-based time advance filter was introduced to remedy the frequency resolution problems of the STFT filter in the previous chapter. This new filter allowed for very accurate time advancement of non-random signals. Although the algorithm showed excellent performance, sources of error were identified and quantified. These sources of error were related to small tonal separation and transients. The window size parameter was shown to play a key role in limiting the error from signals with these characteristics. Knowledge of the disturbance signal characteristics was shown to be important in determining this parameter. Also, the performance of different window types was analyzed and the Hann window was chosen as best for this application.

## CHAPTER 6. INTERNAL MODEL CONTROL

## 6.1 Introduction

This chapter will demonstrate active noise control performance using the previously developed time advance filter. Both a standard feedback controller and an internal model controller typically used for systems with delays will be investigated. Controller performance will be improved with the implementation of a periodic signal selector which separates periodic noise from random noise in real-time. An analysis of the internal model controller stability will be given with conditions derived from the Nyquist stability criteria. Closed loop experimental results of the control system using several example noises will be shown using spectrograms of both the uncontrolled and controlled signals.

## 6.2 Feedback Active Noise Control

Now that the time advance filter has been successfully demonstrated, it can be used as the primary mechanism in the controller. A very basic feedback scheme can be implemented which takes advantage of the time advancement. The block diagram of Fig. 6.1 shows such a configuration. Here, the disturbance noise, d, is recorded by a microphone at the summing junction along with the output of the control speaker, u, after it has traveled through the plant dynamics. This microphone signal is denoted as the control loop error, e, and is fed back through the controller. The controller acts



Figure 6.1 Feedback control block diagram.

on the error with the time advance filter,  $e^{\tilde{t}_d s}$ , and inverse system dynamics,  $\overline{G_{mp}(s)}^{-1}$ , scaled by a constant gain K. The result is then negated and the controller output, u, is played by the control speaker. This signal undergoes the acoustic dynamics of the plant and is recorded once again by the microphone along with the disturbance noise.

The nonminimum phase plant can be broken up into its minimum phase plant,  $G_{mp}$ , and its time delay,  $t_d$ , using the decomposition technique outlined in chapter 3. Inside the controller, the time advance filter effectively eliminates the time delay and the minimum phase plant is inverted to neutralize the dynamics of the system. Eq. 6.1 shows the closed loop transfer function of the configuration of Fig. 6.1. Examining Eq. 6.1, if  $G_{mp}(s) = \overline{G_{mp}(s)}$  and  $t_d = \tilde{t_d}$ , then perfect control should be possible. The closed loop transfer function would reduce to  $\frac{1}{1+K}$  and performance would theoretically be limited only by speaker constraints which limit the size of K.

$$\frac{E(s)}{D(s)} = \frac{1}{1 + KG_{mp}(s)\overline{G_{mp}(s)}^{-1}e^{(\widetilde{t_d} - t_d)s}}$$
(6.1)

The controller of Fig. 6.1 was implemented and the performance results are given



Figure 6.2 Standard feedback configuration yields oscillation in performance due to control delays.

in Fig. 6.2. For this experiment the control speaker and feedback microphone were separated by about 1 meter, a 300 Hz sine wave was used as the disturbance noise, and the gain was set at K = 2. Notice that the controller was not turned on and off but was on from the start, so the oscillations that occur were induced by the controller. Reducing K still produces the same effect, and the inversion of the minimum phase plant must be stable since all poles and zeros are located in the left half plane. The problem must then be within the time advance filter.

The cause of the oscillation must stem from the assumption that  $e^{\tilde{t}_d s}$  advances the signal in time. In actuality, the signal only appears to be advanced in time because the periodic components of the signal have been phase shifted by the appropriate amount. To achieve this phase shift using the phase vocoder, the signal must be electronically delayed by  $(N + hop)/f_s$  seconds. This dead time must be causing the feedback loop to

not behave as expected. In fact, one can visualize how this oscillation occurs by tracing the path of a signal as it travels a few cycles around the feedback loop.

Following along with Fig. 6.1, one can start by imagining a large sinusoidal disturbance signal first entering into the system. The signal is delayed and phase adjusted by the time advance filter and passes through the negated inverse dynamics of the minimum phase plant before it is output through the control speaker. The signal passes through the air, undergoing the acoustic dynamics and time delay before it is recorded by the feedback microphone. At the microphone, the control signal and the disturbance signal will cancel each other quite well as long as the disturbance hasn't changed much. The resulting error signal is therefore small. Consequently, the controller output will be small and so then will the signal entering the microphone to cancel the disturbance. This presents a problem since the disturbance is much larger than the control signal. Therefore, the error at the microphone will be large once again. This cycle repeats itself at regular intervals in accordance with the dead time introduced by the time advance filter (see Fig. 6.2).

### 6.3 Internal Model Control

Attempting to remedy the oscillating performance characteristic of the feedback control scheme of Fig. 6.1, a modification to the controller is presented in Fig. 6.3. This modified control scheme is known as internal model control (IMC) since it mimics the effects of the plant inside the controller using a plant model. IMC is known to provide superior control to plants with inherent time delays [17]. The control signal travels through parallel channels, one in the physical world and one inside the controller. The difference between the two are then calculated. If the plant and the plant model match



Figure 6.3 Internal model control block diagram.

perfectly  $(G_{mp}(s)e^{t_ds} = \overline{G_{mp}(s)}e^{\overline{t_d}s})$ , then the input into the time advance filter,  $e^{\widetilde{t_d}s}$ , becomes the disturbance signal, d. This is exactly what is needed to produce a control signal which will cancel the disturbance at the microphone.

The closed loop transfer function of the IMC configuration is given in Eq. 6.2. This equation differs greatly from the previous closed loop transfer function since it does not rely on a high gain, K, to achieve performance. Instead this loop depends solely on how well the actual plant matches the plant model and how accurately the time advance filter can match the actual time delay in the system. If all of these match perfectly (i.e.  $G_{mp}(s) = \overline{G_{mp}(s)}$  and  $t_d = \overline{t_d} = \widetilde{t_d}$ ), then  $\frac{E(s)}{D(s)} = \frac{0}{1}$ .

$$\frac{E(s)}{D(s)} = \frac{1 - e^{(\widetilde{t_d} - \overline{t_d})s}}{1 - e^{(\widetilde{t_d} - \overline{t_d})s} + G_{mp}(s)\overline{G_{mp}(s)}^{-1}e^{(\widetilde{t_d} - t_d)s}}$$
(6.2)

As a test of the internal model controller performance, the same 300 Hz sine wave



Figure 6.4 IMC solves control oscillation problem and reduces noise of 300 Hz test signal substantially.

was used as a disturbance signal. The resulting signal from the feedback microphone was plotted in Fig. 6.4. Like earlier, the controller was on for the entire duration but does not start reducing the noise until 0.25 seconds due to the delay required by the time advance filter. From the figure it appears the oscillation problem of earlier was fixed with the introduction of the IMC. After a short ramping down of the error (caused by the overlapping windows of the ISTFTs in the time advance filter), the microphone senses almost no noise. The sinusoid disturbance is almost entirely gone and much of what remains is random noise which is uncontrollable with the time advance filter.

#### 6.3.1 Periodic Signal Selector

With the current approach, the entire disturbance signal is processed. The time advance filter phase advances all frequencies regardless of whether the frequency contains



Figure 6.5 Using the IMC, tonal noise reduction is good, but surrounding broadband noise is increased.

periodic or random noise. Advancing the phase of random (broadband) noise is problematic because the time advance filter operates under the assumption the input signal is periodic (tonal). When random noise is sent into the controller, the signal is advanced in phase assuming the disturbance will behave in a predictable manner. However, since the disturbance is unpredictable, the controller output is not capable of canceling the noise. The result is that this phase advanced random noise is added to the disturbance random noise at the microphone, increasing the overall broadband noise.

The negative effect of broadband noise on control performance can be seen in Fig. 6.5. Here, the disturbance was made up of a 200 Hz tone set amidst a moderate level of broadband random noise. With the controller on, the tone was reduced substantially but the broadband noise was increased by 1-2 dB on average. Since many real world disturbance noises contain substantial amounts of broadband noise, this undesirable result

seems very troublesome. Thankfully there are many ways to overcome this problem, one of which will be described next.

The periodic signal selector developed here does just as its name suggests. It removes the random portions from the periodic portions of an incoming signal so that only the periodic parts are left to output. A simple method which extracts tonal elements of a signal can be accomplished by manipulating the magnitude spectrum of the STFT of the signal. In the time advance filter, the phase vector was manipulated but the magnitude vector was passed through unchanged. The periodic signal selector can be placed within the time advance filter as shown in Fig. 6.6. Here, the magnitude vector is analyzed and truncated so that only frequencies containing tones are passed through. The selection procedure is given by Eq. 6.3.

$$A_t(\omega)^* = \begin{cases} A(\omega) & \text{if} \quad \frac{A(\omega)}{\sum_{n=-k}^k A(\omega - n\Delta\omega)} \ge thresh\\ 0 & \text{if} \quad \frac{A(\omega)}{\sum_{n=-k}^k A(\omega - n\Delta\omega)} < thresh \end{cases}$$
(6.3)

for each  $\omega$ , where  $\Delta \omega$  is the frequency spacing.

In Eq. 6.3,  $A_t(\omega)$  is the magnitude vector and  $A_t(\omega)^*$  is the truncated magnitude vector. Essentially the inequality works such that if the magnitude at a center frequency bin is much larger than the magnitude of its neighboring frequencies, then the center frequency magnitude is retained. If, however, the center frequency bin magnitude is not much larger than the surrounding magnitudes, then the center frequency magnitude is set to zero. This works to isolate the tonal frequencies since tones show up as large spikes in the magnitude spectrum of the FFT.

The parameters thresh and n are determined by the user based on disturbance noise characteristics. For instance, the thresh value would be set high if the tonal portion



Figure 6.6 The STFT magnitude vector of the time advance filter is truncated by the periodic signal selector to lessen random noise effects on performance.

was substantially louder than the surrounding random noise. The k value would be set to a large value if a large tonal separation was observed in the disturbance. If *thresh* and n are chosen correctly,  $A_t(\omega)^*$  should be a sparsely populated vector containing only magnitudes of tonal frequencies.

To see the effect of adding the periodic signal selector to the IMC feedback path, the same 200 Hz tone with moderate broadband noise from Fig. 6.5 was used for direct comparison. The results of this test are shown in Fig. 6.7. The broadband noise which was previously made worse by the controller is now unaffected with the inclusion of the periodic signal selector. Also of note, the 200 Hz tone shows improved reductions by 1-2 dB over the previous test. Exhibiting good performance where desired without increasing noise elsewhere is typically impossible with linear system feedback control due to the waterbed effect [17]. This method is able to achieve excellent performance with little to no side effects since it uses the nonlinear process of FFT filtering inside



Figure 6.7 Using the periodic signal selector within the STFT allows for tonal noise reduction without increasing broadband noise.

the controller.

Another key advantage to the periodic signal selector is that it can be used to set boundaries on the controllable frequencies. The control designer can easily set lower and upper limits of frequencies involved in control simply by assigning zero to the magnitude vector elements associated with those frequencies. This is extremely convenient since the control bandwidth could be precisely chosen without the need for filters. It would also be possible to limit control authority for certain individual frequencies if that were desired. Manipulating the magnitude spectrum of the STFT is akin to having ultimate filtering abilities which are not bound by linear filter constraints such as roll-off and phase shifting.

When deciding upon controller bandwidth, a few considerations must be made. First, we would like to remain in the acoustically modal region of the room. This frequency
upper bound is given by the Schroeder cut-off and is described by Eq. 6.4 [25]. In general, for frequencies above this cut-off, acoustic coherence is lost between speaker and microphone since the modes of the room overlap more and more at higher frequencies. In this equation, c is the speed of sound,  $T_{60}$  is the room reverberation time, and V the room volume. The Schroeder cut-off frequency for the mock tractor cab was determined to be 475 Hz, so the control should be band-limited to below this frequency.

$$f_{Sch} = \left(\frac{c^3}{4ln10}\right)^{\frac{1}{2}} \left(\frac{T_{60}}{V}\right)^{\frac{1}{2}}$$
(6.4)

One must also consider the unavoidable performance degradation that comes with modeling errors. This reduction in performance eventually leads to instability if large enough modeling errors are present. The same modeling error can cause much more performance/stability problems at higher frequencies since the wavelengths are smaller and thus more susceptible to phase errors. For example, if the error microphone is a few cm from the user's ear, the performance would not diminish much at 100 Hz since the position error is so small comparable to the wavelength of the signal. However, at 1000 Hz the wavelength is of similar size to the position error, leading to a much larger phase error. In practice, keeping this phase error small usually involves limiting the controller bandwidth to around 1 kHz. Often times even smaller bandwidth is used to improve performance and/or guarantee stability.

#### 6.3.2 Stability

The first step to understanding the stability of a closed loop system is to determine the loop gain. The loop gain transfer function is obtained by multiplying the plant transfer function by the controller transfer function when the plant and controller are in a negative feedback loop [26]. The plant dynamics are given as  $G_{mp}(s)e^{-t_ds}$ . The controller dynamics can be found by looking at Fig. 6.3 and determining the dynamics from controller input, e, to controller output, u. The negative sign can be omitted from  $\overline{G_{mp}(s)}$  since it is contained within the negative feedback loop. The controller dynamics are given in Eq. 6.5.

$$\frac{U(s)}{E(s)} = \frac{\overline{G_{mp}(s)}^{-1} e^{t_d s}}{1 + e^{(\tilde{t}_d - \overline{t_d})s}}$$
(6.5)

The loop gain can then be found by multiplying the controller dynamics of Eq. 6.5 by the plant dynamics,  $G_{mp}(s)e^{-t_ds}$ . The result is given by Eq. 6.6. This equation is expanded further so the magnitude and phase of the minimum phase plant can each be considered in the condition for stability.

$$L(s) = \frac{G_{mp}(s)\overline{G_{mp}(s)}^{-1}e^{(t_d - t_d)s}}{1 + e^{(t_d - t_d)s}}$$
(6.6)

$$L(j\omega) = \frac{A(\omega)e^{\phi(\omega)}e^{j(\overline{t_d} - t_d)\omega}}{\overline{A(\omega)}e^{j\overline{\phi(\omega)}}(1 + e^{j(\overline{t_d} - \overline{t_d})\omega})}$$

$$=\frac{A(\omega)e^{j(\phi(\omega)-\overline{\phi(\omega)}+(\widetilde{t_d}-t_d)\omega)}}{\overline{A(\omega)}(1+e^{j(\widetilde{t_d}-\overline{t_d})\omega})}$$

It is known from the Nyquist stability criterion that the closed loop becomes unstable when  $L(j\omega)$  encircles -1 in the complex plane [26]. That means the loop is unstable if both  $|L(j\omega)| > 1$  and  $\angle L(j\omega) = -\pi$  for any  $\omega$ . What follows is a derivation of the condition for instability in terms of the unknown variables and their estimates. The result will identify the required accuracy of the time advance and tracking systems to maintain stability. Using the Nyquist magnitude constraint, the magnitude condition of the loop is derived below in terms of the system's unknown variables and estimates.

$$|L(j\omega)| \ge 1 \tag{6.7}$$

$$\left|\frac{A(\omega)e^{j(\phi(\omega)-\overline{\phi(\omega)}+(\widetilde{t_d}-t_d)\omega)}}{\overline{A(\omega)}(1+e^{j(\widetilde{t_d}-\overline{t_d})\omega})}\right| \ge 1$$

$$\left|\frac{A(\omega)}{\overline{A(\omega)}\left(1+\cos(\omega(\tilde{t_d}-\overline{t_d}))+j\sin(\omega(\tilde{t_d}-\overline{t_d}))\right)}\right| \ge 1$$

$$\frac{A(\omega)}{\overline{A(\omega)}\sqrt{\left(1+\cos(\omega(\tilde{t_d}-\overline{t_d}))\right)^2+\sin^2(\omega(\tilde{t_d}-\overline{t_d}))}} \ge 1$$

$$\frac{A(\omega)}{\overline{A(\omega)}\sqrt{2+2\cos\left(\omega(\tilde{t_d}-\overline{t_d})\right)}} \ge 1$$

$$\frac{A(\omega)}{2\overline{A(\omega)}\sqrt{\cos^2\left(\frac{\omega}{2}(\tilde{t_d}-\overline{t_d})\right)}} \ge 1$$

$$\frac{A(\omega)}{2\overline{A(\omega)} \Big| \cos\left(\frac{\omega}{2}(\tilde{t_d} - \overline{t_d})\right) \Big|} \ge 1$$

$$\left|\cos\left(\frac{\omega}{2}(\tilde{t}_d - \overline{t}_d)\right)\right| \le \frac{A(\omega)}{2\overline{A(\omega)}} \tag{6.8}$$

Eq. 6.8 is the condition derived from the Nyquist stability gain constraint. This condition by itself does not identify whether or not the closed loop system is stable. Both the gain constraint and the phase constraint must be satisfied at some  $\omega$  for instability to occur. Using the Nyquist phase constraint, the phase condition of the loop is derived below in terms of the system's unknown variables and estimates.

$$\angle L(j\omega) = -\pi \pm 2\pi n \text{ where } n \in \mathbf{Z}$$
(6.9)

$$\phi(\omega) - \overline{\phi(\omega)} + \omega(\tilde{t}_d - t_d) - \frac{\omega}{2}(\tilde{t}_d - \overline{t}_d) = -\pi \pm 2\pi n \text{ where } n \in \mathbf{Z}$$
(6.10)

Now that both the magnitude condition of Eq. 6.8 and the phase condition of Eq. 6.10 have been derived, they can be combined into the complete stability criterion given as Eq. 6.11.

The closed loop system is unstable if for some  $\omega$ :

$$\begin{cases} \left|\cos\left(\frac{\omega}{2}(\tilde{t}_d - \overline{t}_d)\right)\right| \le \frac{A(\omega)}{2A(\omega)} \\ \phi(\omega) - \overline{\phi(\omega)} + \omega(\tilde{t}_d - t_d) - \frac{\omega}{2}(\tilde{t}_d - \overline{t}_d) = -\pi \pm 2\pi n \text{ where } n \in \mathbf{Z} \end{cases}$$
(6.11)

The system will be unstable if both conditions of Eq. 6.11 are met for some  $\omega$ . In an attempt to simplify, lets assume the modeled acoustic system is equivalent to the real acoustic system. This would mean that  $A(\omega)e^{j\phi(\omega)} = \overline{A(\omega)}e^{j\overline{\phi(\omega)}}$  and Eq. 6.11 could be simplified to Eq. 6.12.

$$\begin{cases} \left| \cos\left(\frac{\omega}{2}(\tilde{t}_d - \bar{t}_d)\right) \right| \leq \frac{1}{2} \\ \omega(\tilde{t}_d - t_d) - \frac{\omega}{2}(\tilde{t}_d - \bar{t}_d) = -\pi \pm 2\pi n \text{ where } n \in \mathbf{Z} \end{cases}$$
(6.12)

At the threshold of instability with respect to the magnitude condition (i.e. when  $\left|\cos\left(\frac{\omega}{2}(\tilde{t_d}-\bar{t_d})\right)\right|=\frac{1}{2}$ ):

$$\begin{cases} \omega(\tilde{t}_d - \overline{t}_d) = \pm \frac{2\pi}{3} + 2\pi n\\ \omega(\tilde{t}_d - t_d) = \mp \frac{2\pi}{3} + 2\pi n\\ \omega(\overline{t}_d - t_d) = \pm \frac{2\pi}{3} + 2\pi n \end{cases}$$
(6.13)

Note the third expression of Eq. 6.13 is obtained by rearranging the phase condition of Eq. 6.12 to the equivalent  $\omega(\overline{t_d}-t_d)+\frac{\omega}{2}(\tilde{t_d}-\overline{t_d})=-\pi$ . From these three expressions, a point on the stability boundary can be obtained. If we let  $f_{max}$  be considered the highest controlled frequency, the boundary of instability becomes that of Eq. 6.14. Note  $f_{max}$ can easily be assigned by the control engineer by setting an upper frequency boundary within the periodic signal selector.

Closed loop stability boundary at 
$$\begin{vmatrix} \tilde{t_d} - \overline{t_d} \\ \tilde{t_d} - t_d \\ \overline{t_d} - t_d \end{vmatrix} = \frac{1}{3f_{max}}$$
 (6.14)

At the worst case scenario with respect to the magnitude condition (i.e. when  $\left|\cos\left(\frac{\omega}{2}(\tilde{t}_d - \bar{t}_d)\right)\right| = 0$ ):

$$\begin{cases} \omega(\tilde{t}_d - \bar{t}_d) = \pm \pi + 2\pi n\\ \omega(\tilde{t}_d - t_d) = \mp \frac{\pi}{2} + 2\pi n\\ \omega(\bar{t}_d - t_d) = \pm \frac{\pi}{2} + 2\pi n \end{cases}$$
(6.15)

From these three expressions, another stability boundary point can be obtained (Eq. 6.16).

Closed loop stability boundary at :

$$\left| \tilde{t_d} - \overline{t_d} \right| = \frac{1}{2f_{max}} \quad \& \quad \left| \begin{array}{c} \overline{t_d} - t_d \\ \overline{t_d} - t_d \end{array} \right| = \frac{1}{4f_{max}}$$
(6.16)

The stability boundary as it relates to the various delay estimation errors can be pieced together using all of these stability conditions. Fig. 6.8 shows the region of stability and instability as it pertains to the delay estimation errors.  $|\tilde{t_d} - \bar{t_d}|$  is an error which is solely the consequence of errors within the time advance filter (random noise, tonal separation, chirps/transients).  $|\bar{t_d} - t_d|$  and  $|\tilde{t_d} - t_d|$  are errors which are largely dependent on how the measured time delay differs from the actual delay and is therefore caused by user position tracking errors.

The key thing to notice about Fig. 6.8 is that as long as the time advance filter error is small, the system will remain stable even if the tracking system is poor. Likewise, as long as the tracking error is small, the system will remain stable even if the time advance filter malfunctions. The system is therefore quite robust since instability can only result if both the tracking system and the controller exhibit large error concurrently.

For further stability analysis, the previous assumption that  $A(\omega)e^{j\phi(\omega)} = \overline{A(\omega)}e^{j\overline{\phi(\omega)}}$ is relaxed so that the effect of error in the modeled acoustics can be observed. In Fig. 6.9, the stability region is recalculated assuming the modeled acoustics have 6 dB magnitude error and/or 90° phase error. As expected, the plots show the region of stability shrinking with increased error.

It seems clear looking at these plots that  $|\overline{t_d} - t_d|$  should probably be kept under  $\frac{1}{4f_{max}}$  to ensure stability even with multiple sources of error present. If, for example, 400 Hz



Figure 6.8 Region of instability shown as a function of time advance filter error (y-axis) and tracking error (x-axis).



Figure 6.9 Region of stability diminishes with increasing error between modeled and actual plant.

was the desired maximum controlled frequency, this would mean the difference between the actual and measured time delays could be no larger than 1/1600 sec. This may seem small, but this time corresponds to a tracking error of 21.4 cm. Keeping error within this range should be no problem for the ultrasonic tracking system described in chapter 2. Of course one would hope for better tracking errors than those near the boundary of instability, since controller performance would most certainly be poor there.

#### 6.4 Experimental Results

To test the performance of the IMC closed loop system, a wide variety of sound files were obtained from www.freesound.org for use as disturbances [27]. The website has millions of files in a searchable database that is free to anyone as long as proper credit is given. This provided a quick way of testing the controller with noises containing diverse spectrums without spending time in the field recording them personally. Using the website has the drawback of one not being able to get exactly what one wants sometimes. There are clips which do not have very good descriptions, there are some corrupted with noise, and there are some that are just a few seconds in length and therefore too short to show controller performance. Despite the limitations, many decent audio files were found of real world disturbances. This allows for a more convincing test of the controller's capabilities than if only computer generated sound files were used.

Using the free sound database provided a quick and easy way of determining how the controller *might* work in a variety of situations. The examples presented are therefore not actual results from on-site measurements but are simulated results using recordings of the disturbances. The experimental setup for all of the examples was the same. All of the disturbance noises were played through a speaker outside of a small room and

recorded by a microphone inside the room. The control speaker was also inside the room (refer to chapter 3).

A description of each result will be given here so that each set of before and after plots can each be shown on the same page for direct comparison. Many of the plots are very low frequency tonal noise since that is where the IMC controller's strength lies. Each sound file was bandpass filtered and re-sampled at 1024 Hz since only frequencies between 25-500 Hz were considered for control.

• Water Pump [28]

Fig. 6.10 and Fig. 6.11 show excellent reduction of the primary tone and two harmonics. The noise reduction of the tones is almost down to the surrounding broadband noise.

• Desk Fan [29]

Fig. 6.12 and Fig. 6.13 show excellent reduction in the primary tone with small reductions in the harmonics. The close proximity of the tones provided a challenge to the controller.

• Bathroom Fan [30]

Fig. 6.14 and Fig. 6.15 show moderate reduction of the primary and secondary tones. Note the large amount of higher frequency noise which was ignored by the controller.

• Inside a Moving Bus [31]

Fig. 6.16 and Fig. 6.17 show very good reductions to almost the level of the surrounding broadband noise.

• Idling Bobcat [32]

Fig. 6.18 and Fig. 6.19 show very good reductions of the two primary tones. Several tones around 200 Hz are not reduced as much due to their close spacing.

• Diesel Engine [33]

Fig. 6.20 and Fig. 6.21 show somewhat spotty performance, perhaps due to the transient explosions in the engine. The engine is revved 8 seconds into the recording. The controller neither reduces, nor adds to the noise much here.

• On Large Boat [34]

Fig. 6.22 and Fig. 6.23 show moderate performance using a large FFT window to reduce the very closely spaced tones.

• Ship Cabin [35]

Fig. 6.24 and Fig. 6.25 show moderate reductions of the few low frequency tones which are varying with intensity.

• Inside Driving Car [36]

Fig. 6.26 and Fig. 6.27 show poor reductions of rapidly changing frequencies. The control can't reduce constant frequencies since the window size is set very low to accommodate fast moving tones.

• Inside Car Getting onto Highway [37]

Fig. 6.28 and Fig. 6.29 show poor reductions of rapidly changing frequencies. Slight noise reduction is almost unnoticeable with so much random noise dominating the disturbance.

• Air Compressor [38]

Fig. 6.30 and Fig. 6.31 show excellent performance at multiple tones. The tone

at 100 Hz is only slightly reduced due to its varying intensity.

• Inside Small Propeller Airplane [39]

Fig. 6.32 and Fig. 6.33 show excellent performance at multiple tones. Performance is surprising due to the pulsating frequency behavior.



Figure 6.10 Spectrogram of water pump.



Figure 6.11 Spectrogram of water pump with IMC.

Spectrogram of Desk Fan



Figure 6.12 Spectrogram of desk fan.



Spectrogram of Desk Fan with IMC

Figure 6.13 Spectrogram of desk fan with IMC.



Figure 6.14 Spectrogram of bathroom fan.



Figure 6.15 Spectrogram of bathroom fan with IMC.

110



Figure 6.16 Spectrogram of bus.



Figure 6.17 Spectrogram of bus with IMC.



Figure 6.18 Spectrogram of Bobcat idling.



Figure 6.19 Spectrogram of Bobcat idling with IMC.



Figure 6.20 Spectrogram of diesel engine.



Figure 6.21 Spectrogram of diesel engine with IMC.







Spectrogram of Large Boat with IMC

Figure 6.23 Spectrogram of large boat with IMC.

Spectrogram of Ship Cabin







Spectrogram of Ship Cabin with IMC

Figure 6.25 Spectrogram of ship cabin with IMC.



Figure 6.26 Spectrogram of car driving.



Figure 6.27 Spectrogram of car driving with IMC.





Figure 6.28 Spectrogram of car getting onto highway.



Spectrogram of Car getting on Highway with IMC

Figure 6.29 Spectrogram of car getting onto highway with IMC.







Figure 6.31 Spectrogram of compressor with IMC.



Figure 6.32 Spectrogram of small propeller airplane.



Figure 6.33 Spectrogram of small propeller airplane with IMC.

### 6.5 Remarks

The performance of a standard feedback controller using the time advance filter showed oscillations stemming from the delay within the system. An internal model controller (IMC) was developed to combat this problem. The IMC combined with the time advance filter showed great tonal noise reductions. With the added periodic signal selector algorithm, tonal noise could be diminished without increasing the surrounding broadband noise. The stability conditions of the closed loop system were derived using the Nyquist stability criterion. The instability boundary was also analyzed by varying the effect of potential sources of error. The performance of the entire system was shown with spectrograms of many example disturbance noises before and after control. The experiments showed large reductions in low frequency tonal noise.

## CHAPTER 7. CONCLUSION AND FUTURE WORK

### 7.1 Conclusion

In chapter 2, an ultrasonic positioning system was developed to track a microphone in 3D space. The microphone position information could be fed into an active noise control system to update internal models and improve performance. The use of continuous bandlimited white noise and time separated bandlimited pulses were both explored. Pulses were deemed superior due to the increased bandwidth available. Both time and frequency domain methods for determining acoustic time delays were given. Many sources of error and causes of incorrect solutions were identified as well as ideas for reducing the error and likelihood of an incorrect solution. Experimental results showed that ultrasonic tracking could be quite accurate for tracking situations involving slow moving targets in a confined space. Ultrasonic tracking also showed its ability to track a target positioned behind an obstruction, a key advantage over infrared tracking.

In chapter 3, the 3D active noise control time delay problem was shown to be the cause of performance and bandwidth limitations because it introduced RHP zeros into the plant dynamics. Non-minimum phase decomposition was introduced as a means of circumventing this problem and was used as a tool for singling out the problematic time delay. The real cepstrum method proved to be the best decomposition technique. A method for dealing with the identified time delay is left to the next chapter. A

frequency response was taken of an example 3D acoustic enclosure which was successfully decomposed and accurately modeled with no RHP zeros.

The idea of a time advance filter was proposed in chapter 4 as a way of combating the time delay problems inherent to 3D noise control. In theory, the successful implementation of such a filter would offer the same excellent closed loop performance as a collocated noise controller. The procedure for how to implement such a filter in the frequency domain was presented with detailed information about deconstructing and accurately reconstructing a signal. The STFT time advancement filter was shown to produce accurate time advanced signals only for a finite number of frequencies corresponding to the FFT center frequency bins. Since this would fail with most real world disturbances, more work needs to be done to make time advancement accurate for all frequencies. This work is done in the next chapter.

The phase vocoder-based time advance filter was introduced in chapter 5 to remedy the frequency resolution problems of the STFT filter in the previous chapter. This new filter allowed for very accurate time advancement of non-random signals. Although the algorithm showed excellent performance, sources of error were identified and quantified. These sources of error were related to small tonal separation and transients. The window size parameter was shown to play a key role in limiting the error from signals with these characteristics. Knowledge of the disturbance signal characteristics was shown to be important in determining this parameter. Also, the performance of different window types was analyzed and the Hann window was chosen as best for this application.

The performance of a standard feedback controller using the time advance filter showed oscillations stemming from the delay within the system. An internal model controller (IMC) was developed in chapter 6 to combat this problem. The IMC combined with the time advance filter showed great tonal noise reductions. With the added periodic signal selector algorithm, tonal noise could be diminished without increasing the surrounding broadband noise. The stability conditions of the closed loop system were derived using the Nyquist stability criterion. The instability boundary was also analyzed by varying the effect of potential sources of error. The performance of the entire system was shown with spectrograms of many example disturbance noises before and after control. The experiments showed large reductions in low frequency tonal noise.

#### 7.2 Future Work

An adaptive plant selector is needed to integrate the ultrasonic tracking system with the control system as shown in Fig. 7.1. This paper shows the performance of each separately, but does not investigate controller performance with a moving target. It is recommended that in the design stage the engineer experimentally identifies a multitude of plant models at known locations within the room. This would create a 3D acoustic map of the room which would be unique to each environment. In this way, all of the potential models would be stored in memory and an adaptation algorithm could be developed to switch and/or interpolate between them.

**Feedforward control** could be added to the design as shown in Fig. 7.2. For the special case in which the disturbance noise source location is fixed, a nearby microphone could be used to preview the noise so that a feedforward controller could provide the appropriate canceling noise. For these types of systems, feedforward control is desirable since it is capable of reducing broadband noise. When combined with feedback control, both periodic and random disturbances can be attenuated.

Linear frequency extrapolation could be implemented to reduce the time advance filter error for quickly varying chirp disturbances. Since the change in phase from one



Figure 7.1 The plant selector is the link between the tracking system and controller.



Figure 7.2 Possible feedforward/feedback integrated controller.

STFT to another identifies the primary frequencies of the disturbance (using the phase vocoder), the change in the change of phase from three STFTs should determine how the frequencies vary with time. With this information, one can extrapolate where the frequency will be in the future. This estimate would likely be more accurate than the current approach of assuming an unchanged frequency.

A multi-resolution STFT approach could also be used to improve performance when a disturbance consists of a very low frequency fundamental tone with several higher frequency harmonics. Since it was shown that a frequency can be accurately identified in just 1-2 wavelengths using the phase vocoder, more time is needed to identify lower frequencies than higher frequencies. Therefore, it would be beneficial to compute multiple STFTs within the same time interval so that higher frequencies could be detected and controlled quicker. Such a system is illustrated in Fig. 7.3, where each block represents a single STFT computation. In this example, the highest frequencies are identified eight times in the same time it takes to identify the lowest frequencies.

The work presented here was designed with active noise control in mind but the functionality of this new type of frequency domain feedback control can extend to other applications. It is the hope of the author that this work is used and built upon in fields which, like 3D ANC, require control of periodic signals and are limited by a large actuator to sensor delay.



Figure 7.3 STFT computation blocks of a multi-resolution system.

# BIBLIOGRAPHY

- Kuo, Sen M., Morgan, Dennis R., "Active Noise Control: A Tutorial Review." Proceedings of the IEEE, Vol. 87, No. 6, Jun 1999.
- [2] Proakis, John G., Manolakis, Dimitris G., Digital Signal Processing: Principles, Algorithms, and Applications, 4th Edition. Prentice Hall, Upper Saddle River, New Jersey, 2007.
- [3] Das, D.P., Panda, G., Kuo, S.M., "New Block Filtered-X LMS Algorithms for Active Noise Control Systems." IET Signal Processing, Vol. 1, No. 2, 2007.
- [4] Morgan, Dennis R., Thi, James C., "A Delayless Subband Adaptive Filter Architecture." IEEE Transactions on Signal Processing, Vol. 43, No. 8, Aug 1995
- [5] Kuo, Sen M., Chung, Kai M., "Narrowband Active Noise Control Using Adaptive Delay Filter." IEEE Signal Processing Letters, Vol. 5, No. 12, Dec 1998.
- [6] Puckette, Miller S., Brown, Judith C., "Accuracy of Frequency Estimates using the Phase Vocoder." IEEE Transactions on Speech and Audio Processing, Vol. 6, No. 2, Mar 1998.
- [7] Nelson, P.A., Elliott, S.J., Active Control of Sound. Academic Press, New York 1992.

- [8] Rafaely, Boaz, "Zones of Quiet in a Broadband Diffuse Sound Field." Acoustical Society of America, 2001.
- [9] Guo, Jingnan, Pan, Jie, "Actively Created Quiet Zones for Broadband Noise using Multiple Control Sources and Error Microphones." Acoustical Society of America, 1999.
- [10] Copley, David C., Faber, Ben, "Energy Density Active Noise Control in an Earthmoving Machine Cab." Noise-Con, Minneapolis, Minnesota, Oct 2005.
- [11] Gonzalez, J.R., Bleakley, C.J., "High Precision Robust Broadband Ultrasonic Location and Orientation Estimation." IEEE Journal of Selected Topics in Signal Processing, Vol. 3, No. 5, Oct 2009.
- [12] Dijk, Esko O., "Indoor Ultrasonic Position Estimation Using a Single Base Station." PhD Thesis, Technische Universiteit Eindhoven, 2004.
- [13] Lee, Johnny Chung, "Hacking the Nintendo Wii Remote." Pervasive Computing, Jul-Sep 2008.
- [14] Kirchner, Nathan, Furukawa, Tomonari, "Infrared Localisation for Indoor UAVs." 1st International Conference on Sensing Technology, Palmerston North, New Zealand, Nov 2005.
- [15] Mungamuru, Bob, Aarabi, Parham, "Enhanced Sound Localization." IEEE Transactions on Systems, Man, and Cybernetics, Vol. 34, No. 3, Jun 2004.
- [16] Kwon, Byoungho, Park, Youngjin, Park, Youn-sik. "Sound Source Localization for Robot Auditory System Using the Summed GCC Method." International Conference on Control, Automation and Systems, Seoul, Korea, Oct 2008.

- [17] Skogestad, Sigurd, Postlethwaite, Ian, Multivariable Feedback Control. John Wiley and Sons Inc., West Sussex, England 1996.
- [18] Oppenheim, A.V., Schafer, R.W., Discrete-Time Signal Processing. Englewood Cliffs, New Jersey, Prentice Hall, 1989.
- [19] Al-Bassyiouni, Moustafa, "Active Control of Sound Transmission into Three-Dimensional Enclosures." PhD Thesis, University of Maryland, 2004.
- [20] Juang, Jer-Nan, Applied System Identification. Prentice Hall Inc., Englewood Cliffs, New Jersey 1994.
- [21] Waite, T.C., Kelkar, Atul, "System Identification using Enhanced SOCIT Algorithms," International Conference on Advances in Control and Optimization of Dynamical Systems, Bangalore, India, Feb 1, 2007.
- [22] Lathi, B.P., Signal Processing and Linear Systems. Berkeley-Cambridge Press, Carmichael, California 1998.
- [23] Smith, Julius O., Spectral Audio Signal Processing. March 2010 Draft. https://ccrma.stanford.edu/ jos/sasp/.
- [24] Sethares, William A., Rhythm and Transforms. Springer-Verlag London Limited 2007.
- [25] Pierce, Allan D., Acoustics: An Introduction to its Physical Principles and Applications. Acoustical Society of America, Melville, New York 1989.
- [26] Franklin, Gene F., Powell, J. David, Emami-Naeini, Abbas, Feedback Control of Dynamic Systems, 4th Edition. Prentice Hall, Upper Saddle River, New Jersey 2002.

- [27] de Jong, Bram. The Freesound Project. Web. jhttp://www.freesound.orgj.
- [28] luffy. "luffy water5.wav". The Freesound Project. Web. Apr 2, 2006.
- [29] fresco. "desk ventilator by fresco.wav". The Freesound Project. Web. Jun 3, 2007.
- [30] RHumphries. "RBH Household Bathroom fan.wav". The Freesound Project. Web. Apr 18, 2005.
- [31] Koops. "Bus Engine 01 LOOP3.wav". The Freesound Project. Web. Dec 9, 2005.
- [32] BoilingSand. "Bobcat idling 080320.wav". The Freesound Project. Web. Mar 26, 2008.
- [33] dobroide. "20060329.diesel.engine.flac". The Freesound Project. Web. Apr 2, 2006.
- [34] Anton. "Big Boat.wav". The Freesound Project. Web. Aug 2, 2005.
- [35] inchadney. "Shipscabin.WAV". The Freesound Project. Web. Nov 8, 2009.
- [36] hanstimm. "cardrive 2.mp3". The Freesound Project. Web. Oct 22, 2007.
- [37] Heigh-hoo. "getting on the highway.aiff". The Freesound Project. Web. Nov 15, 2006.
- [38] andysm. "R09 0005B.wav". The Freesound Project. Web. Jul 30, 2009.
- [39] sebastianlund. "Airplane Small Propeller Take off.WAV". The Freesound Project. Web. Dec 21, 2007.